МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

H.222.0

(05/2006)

СЕРИЯ H: АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

Инфраструктура аудиовизуальных служб – Мультиплексирование и синхронизация при передаче

Информационная технология – Общее кодирование подвижных изображений и соответствующей аудиоинформации: Системы

Рекомендация МСЭ-Т H.222.0

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ Н

АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

ХАРАКТЕРИСТИКИ ВИДЕОТЕЛЕФОННЫХ СИСТЕМ	Н.100–Н.199
ИНФРАСТРУКТУРА АУДИОВИЗУАЛЬНЫХ УСЛУГ	
Общие положения	Н.200–Н.219
Мультиплексирование и синхронизация при передаче	**Н.220–Н.229**
Системные аспекты	Н.230–Н.239
Процедуры связи	Н.240–Н.259
Кодирование движущихся видеоизображений	Н.260–Н.279
Сопутствующие системные аспекты	Н.280–Н.299
Системы и оконечное оборудование для аудиовизуальных услуг	Н.300–Н.349
Архитектура услуг справочника для аудиовизуальных и мультимедийных услуг	Н.350–Н.359
Качество архитектуры обслуживания для аудиовизуальных и мультимедийных услуг	Н.360–Н.369
Дополнительные услуги для мультимедиа	Н.450–Н.499
ПРОЦЕДУРЫ МОБИЛЬНОСТИ И СОВМЕСТНОЙ РАБОТЫ	
Обзор мобильности и совместной работы, определений, протоколов и процедур	Н.500–Н.509
Мобильность для мультимедийных систем и услуг серии Н	Н.510–Н.519
Приложения и услуги мобильной мультимедийной совместной работы	Н.520–Н.529
Защита мобильных мультимедийных систем и услуг	Н.530–Н.539
Защита приложений и услуг мобильной мультимедийной совместной работы	Н.540–Н.549
Процедуры мобильного взаимодействия	Н.550–Н.559
Процедуры взаимодействия мобильной мультимедийной совместной работы	Н.560–Н.569
ШИРОКОПОЛОСНЫЕ И МУЛЬТИМЕДИЙНЫЕ TRIPLE-PLAY УСЛУГИ	
Предоставление широкополосных мультимедийных услуг по VDSL	Н.610–Н.619

Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.

МЕЖДУНАРОДНЫЙ СТАНДАРТ ИСО/МЭК 13818-1
РЕКОМЕНДАЦИЯ МСЭ-Т Н.222.0

Информационная технология – Общее кодирование подвижных изображений и соответствующей аудиоинформации: Системы

Резюме

В настоящей Рекомендации | Международном стандарте определяется спецификация системного уровня кодирования. Она была разработана в 1994 году, главным образом, в целях обеспечения поддержки объединения и синхронизации методов кодирования изображения и звука, определенных в Частях 2 и 3 стандарта ИСО/МЭК 13818. С 1994 г. этот стандарт был дополнен с тем, чтобы он поддерживал дополнительные спецификации кодирования изображения (ИСО/МЭК 14496-2 и ИСО/МЭК 14496-10), спецификации кодирования звука (ИСО/МЭК 13818-7 и ИСО/МЭК 14496-3), спецификации кодирования системных потоков (ИСО/МЭК 14496-1 и ИСО/МЭК 15938-1), IPMP (ИСО/МЭК 13818-11), а также общие спецификации кодирования метаданных. Системный уровень поддерживает шесть основных функций:

1) синхронизация множества компрессированных потоков при декодировании;
2) перемежение множества компрессированных потоков в единый поток;
3) запуск процесса буферизации, для того чтобы начать декодирование;
4) непрерывное управление буфером;
5) идентификация времени; и
6) мультиплексирование и передача сигналов различных компонентов системного потока.

Мультиплексированный двоичный поток, соответствующий Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1, представляет собой либо **Транспортный поток,** либо **Программный поток**. Оба потока состоят из **пакетов PES** и пакетов, содержащих другую необходимую информацию. Оба типа потоков поддерживают мультиплексирование компрессированных потоков изображения и звука одной программы с общей временной базой. **Транспортный поток** дополнительно поддерживает мультиплексирование компрессированных потоков изображения и звука нескольких программ с независимыми временными базами. **Программный поток,** как правило, более пригоден для условий работы "почти без ошибок", так как он обеспечивает возможность компьютерной обработки программной информации. Транспортный поток более пригоден для использования в условиях, где вероятно появление ошибок.

Мультиплексированный двоичный поток, соответствующий Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1, вне зависимости от того, является ли он Транспортным или Программным, состоит из двух уровней: внешним уровнем является системный уровень, а внутренним – уровень компрессии. Системный уровень обеспечивает выполнение функций, необходимых для использования в системе одного или нескольких компрессированных потоков данных. Части настоящей спецификации, касающиеся изображения и звука, определяют уровень компрессии кодирования данных изображения и звука. Кодирование других типов данных в настоящей Рекомендации | Международном стандарте не определяется, но оно поддерживается системным уровнем, при условии, что другие типы данных соответствуют ограничениям, определенным в настоящей Рекомендации | Международном стандарте.

Источник

Рекомендация МСЭ-Т Н.222.0 утверждена 29 мая 2003 года. 16-й Исследовательской комиссией МСЭ-Т в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т А.8. Идентичный текст публикуется также в виде стандарта ИСО/МЭК 13818-1.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи и информационно-коммуникационных технологий (ИКТ). Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

На Всемирной ассамблее по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяются темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соблюдение положений данной Рекомендации осуществляется на добровольной основе. Однако данная Рекомендация может содержать некоторые обязательные положения (например, для обеспечения функциональной совместимости или возможности применения), и в таком случае соблюдение Рекомендации достигается при выполнении всех указанных положений. Для выражения требований используются слова "следует", "должен" ("shall") или некоторые другие обязывающие выражения, такие как "обязан" ("must"), а также их отрицательные формы.

1 Употребление таких слов никоим образом не следует интерпретировать как основание для частичного или полного соблюдения положений данной Рекомендации.

2 Употребление таких слов не означает, что от какой-либо стороны требуется соблюдение положений данной Рекомендации.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или выполнение настоящей Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, действительности или применимости заявленных прав интеллектуальной собственности, независимо от того, доказываются ли такие права членами МСЭ или другими сторонами, не относящимися к процессу разработки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения настоящей Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что вышесказанное может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: http://www.itu.int/ITU-T/ipr/.

© ITU 2010

Все права сохранены. Ни одна из частей данной публикации не может быть воспроизведена с помощью каких бы то ни было средств без предварительного письменного разрешения МСЭ.

СОДЕРЖАНИЕ

Стр.

РАЗДЕЛ 1 – ОБЩИЕ ПОЛОЖЕНИЯ 1
1.1 Сфера применения 1
1.2 Нормативные справочные документы 1

РАЗДЕЛ 2 – ТЕХНИЧЕСКИЕ ЭЛЕМЕНТЫ 2
2.1 Определения 2
2.2 Символы и сокращения 6
2.3 Метод описания синтаксиса двоичного потока 8
2.4 Требования к двоичному транспортному потоку 9
2.5 Требования к двоичному программному потоку 54
2.6 Дескрипторы программ и элементов программ 67
2.7 Ограничения семантики мультиплексированного потока 96
2.8 Совместимость с ИСО/МЭК 11172 102
2.9 Регистрация идентификаторов авторских прав 102
2.10 Регистрация формата конфиденциальных данных 102
2.11 Передача данных в соответствии с ИСО/МЭК 14496 103
2.12 Передача метаданных 115
2.13 Передача данных в соответствии с ИСО 15938 125
2.14 Передача видео в соответствии с Рекомендацией МСЭ-Т H.264 | ИСО/МЭК 14496-10 125

Приложение A – Модель декодера CRC 130
A.0 Модель декодера CRC 130

Приложение B – Управление и контроль цифровой среды хранения данных (DSM-CC) 131
B.0 Введение 131
B.1 Общие элементы 132
B.2 Технические элементы 134

Приложение C – Специальная информация программы 140
C.0 Описание информации, относящейся к программе, в Транспортных потоках 140
C.1 Введение 140
C.2 Функциональный механизм 140
C.3 Преобразование секций в пакеты Транспортного потока 141
C.4 Частота повторений и случайный доступ 142
C.5 Что такое программа? 142
C.6 Распределение значений program_number 142
C.7 Использование PSI в типичной системе 143
C.8 Взаимосвязи структур PSI 143
C.9 Использование полосы пропускания и время обнаружения сигнала 146

Приложение D – Модель синхронизации систем и применение положений настоящей Рекомендации | Международного стандарта 148
D.0 Введение 148

Приложение E – Приложения передачи данных 159
E.0 Общие соображения 159
E.1 Предложение 159

Приложение F – Графические обозначения синтаксиса для настоящей Рекомендации | Международного стандарта 160
F.0 Введение 160

Приложение G – Общая информация 164
G.0 Общая информация 164

Приложение H – Конфиденциальная информация 164
H.0 Конфиденциальная информация 164

Приложение I – Соответствие систем и интерфейс реального времени 166
I.0 Соответствие систем и интерфейс реального времени 166

Стр.

Приложение J – Взаимодействие систем, вносящих нестабильность синхронизации, с декодерами MPEG 166
J.0 Введение 166
J.1 Модели соответствия систем 167
J.2 Спецификация сети для уменьшения нестабильности синхронизации 168
J.3 Примеры реализации декодера 169

Приложение K – Объединение транспортных потоков 170
K.0 Введение 170
K.1 Различные типы точек объединения 170
K.2 Характеристики декодера при объединениях 171

Приложение L – Процедура регистрации (см. § 2.9) 172
L.1 Процедура запроса зарегистрированного идентификатора (RID) 172
L.2 Область ответственности органа регистрации 173
L.3 Область ответственности сторон, запрашивающих RID 173
L.4 Апелляционная процедура для отклоненных обращений 174

Приложение M – Форма заявления на регистрацию (см. § 2.9) 174
M.1 Контактная информация об организации, запрашивающей зарегистрированный идентификатор (RID) 174
M.2 Заявление о намерении применить назначенный RID 174
M.3 Дата предположительного применения RID 174
M.4 Уполномоченный представитель 174
M.5 Только для официального использования органом по регистрации 174

Приложение N 175

Приложение O – Процедура регистрации (см. § 2.10) 176
O.1 Процедура запроса RID 176
O.2 Область ответственности органа регистрации 176
O.3 Контактная информация органа регистрации 176
O.4 Область ответственности сторон, запрашивающих RID 176
O.5 Апелляционная процедура для отклоненных обращений 177

Приложение P – Форма заявления на регистрацию 177
P.1 Контактная информация об организации, запрашивающей RID 177
P.2 Запрос конкретного RID 177
P.3 Краткое описание RID, который используется в развертываемой системе 177
P.4 Заявление о намерении применить назначенный RID 177
P.5 Дата предположительного применения RID 177
P.6 Уполномоченный представитель 177
P.7 Для официального использования органом по регистрации 178

Приложение Q – Модели буфера T-STD и P-STD для ADTS, соответствующего ИСО/МЭК 13818-7 178
Q.1 Введение 178
Q.2 Интенсивность утечки из транспортного буфера 178
Q.3 Размер буфера 178
Q.4 Вывод 180

Приложение R – Передача сцен, соответствующих ИСО/МЭК 14496, в потоках, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 181
R.1 Процедура доступа к содержанию для компонентов программы, соответствующих ИСО/МЭК 14496, в Программном потоке 181
R.2 Процедура доступа к содержанию для компонентов программы, соответствующих ИСО/МЭК 14496, в Транспортном потоке 182

Введение

Часть "Системы" настоящей Рекомендации | Международного стандарта касается объединения одного или нескольких элементарных потоков видео и аудио информации, а также других данных в единый поток или в несколько потоков, которые пригодны для хранения или передачи данных. Системное кодирование выполняется в соответствии с синтаксическими и семантическими правилами, установленными настоящей Спецификацией, и предоставляет информацию, позволяющую выполнить синхронное декодирование информации, находящейся в буферах декодеров, в различных условиях получения или приема сигналов.

Системное кодирование будет определено в двух формах: в **Транспортном потоке** и **Программном потоке**. Каждая из них оптимизирована для различных наборов приложений. И Транспортный, и Программный потоки, определенные в настоящей Рекомендации | Международном стандарте, обеспечивают синтаксис кодирования, который является необходимым и достаточным для синхронизации декодирования и воспроизведения видео и аудио информации, гарантируя при этом, что буферы данных в декодерах не переполняются или не оказываются недозаполненными. Информация кодируется с использованием меток времени, необходимых для декодирования и воспроизведения аудио и видеоданных, а также меток времени, необходимых для доставки самого потока данных. Оба потока определены как мультиплексированные потоки на базе пакетов.

Основной метод мультиплексирования для отдельных элементарных видео или аудио потоков показан на рисунке Введение. 1. Видео и аудио данные кодируются, как описано в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 и ИСО/МЭК 13818-3. Результирующие компрессированные элементарные потоки делятся на пакеты, в результате чего создаются **пакеты PES**. При формировании пакетов PES в них может быть добавлена информация, необходимая для того, чтобы пакеты PES можно было использовать независимо от Транспортных или Программных потоков. Эта информация не требуется и не должна добавляться, когда пакеты PES объединяются далее с информацией системного уровня, образуя Транспортные потоки или Программные потоки. Настоящий системный стандарт охватывает процессы, изображенные справа от вертикальной пунктирной линии.

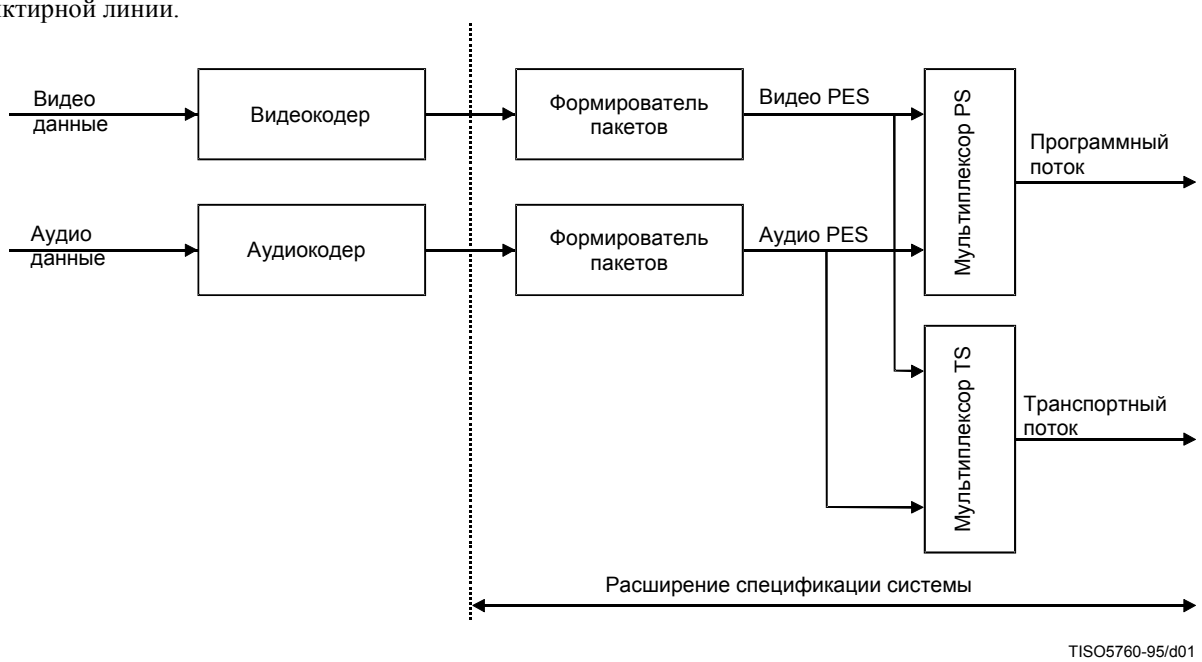


Рисунок Введение. 1 –
Упрощенное изображение сферы действия настоящей Рекомендации | Международного стандарта

Программный поток аналогичен и подобен системному уровню, описанному в ИСО/МЭК 11172. Он является результатом объединения в единый поток одного или нескольких потоков пакетов PES, которые имеют общую временную базу.

Для приложений, в которых требуется, чтобы элементарные потоки, содержащие одну программу, были бы раздельными потоками, которые не мультиплексируются вместе, эти элементарные потоки могут быть также закодированы как отдельные Программные потоки с общей временной базой, по одному на каждый элементарный поток. В таком случае значения, кодированные в виде полей SCR различных потоков, должны быть единообразными.

Точно также, как отдельный Программный поток, все элементарные потоки должны декодироваться синхронно.

Программный поток предназначен для использования в относительно безошибочных условиях и пригоден для таких приложений, которые могут включать в себя программную обработку системной информации, например, для мультимедийных приложений. Пакеты программного потока могут иметь различную и относительно большую длину.

Транспортный поток объединяет в единый поток одну или несколько программ с одной или несколькими независимыми временными базами. Пакеты PES, созданные из элементарных потоков, которые образуют программу, используют общую временную базу. Транспортный поток предназначен для использования в таких условиях, где существует вероятность появления ошибок, например, хранение или передача данных в среде с большими потерями или шумами. Пакеты Транспортного потока имеют длину 188 байтов.

Программные и Транспортные потоки предназначены для различных приложений и их определения не соответствуют в точности многоуровневой модели. Вполне возможно и целесообразно выполнять преобразование одного потока в другой, однако, ни один из них не является составной частью или расширенным вариантом другого. В частности, из Транспортного потока можно выделить содержание программы и создать пригодный Программный поток, это осуществляется при помощи взаимозаменяемых форматов пакетов PES, но в Транспортном потоке содержатся не все поля, необходимые для Программного потока; некоторые из них должны быть получены из имеющихся данных. Транспортный поток может быть использован для того, чтобы охватить некоторый диапазон уровней многоуровневой модели, и предназначен для повышения эффективности и упрощения реализации широкополосных приложений.

Сферы применения синтаксических и семантических правил, установленных в спецификациях систем, различны: синтаксические правила применяются только для спецификаций кодирования системного уровня, и не распространяются на кодирование уровня компрессии для аудио и видео потоков; в отличие от них, семантические правила применяются к комбинированному потоку в целом.

Спецификация систем не определяет архитектуру или вариант реализации ни кодеров или декодеров, ни мультиплексоров или демультиплексоров. Однако свойства двоичного потока устанавливают функциональные требования и требования по качественным показателям для кодеров, декодеров, мультиплексоров и демультиплексоров. Например, кодеры должны удовлетворять требованиям по минимальному допустимому отклонению частоты синхронизации. Несмотря на это и другие требования, при разработке и реализации кодеров, декодеров, мультиплексоров и демультиплексоров предоставляется значительная степень свободы.

Введение. 1 Транспортный поток

Транспортный поток – определяется, как поток, который предназначен для передачи или хранения одной или нескольких программ кодированных данных, соответствующих Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 и ИСО/МЭК 13818-3, а также других данных в условиях, где могут возникать существенные ошибки. Такие ошибки могут проявляться в виде ошибок в значении бита или в виде потери пакетов.

Транспортные потоки могут иметь либо фиксированную, либо переменную скорость передачи. В любом случае, составляющие их элементарные потоки могут иметь либо фиксированную, либо переменную скорость передачи. Ограничения потока по синтаксису и семантике для каждого из этих случаев остаются одними и теми же. Скорость передачи Транспортного потока определяется значениями и местами расположения полей Эталонного времени программы (Program Clock Reference (PCR)), которые, как правило, представляют собой отдельные поля PCR для каждой программы.

Существуют определенные трудности в создании и доставке Транспортного потока, содержащего несколько программ с независимыми временными базами, так как общая скорость передачи такого потока является переменной. См. § 2.4.2.2.

Транспортный поток может быть сформирован любым способом, позволяющим получить пригодный поток. Транспортные потоки, содержащие одну или несколько программ, можно сформировать из элементарных потоков кодированных данных, из Программных потоков или из других Транспортных потоков, которые сами могут содержать одну или несколько программ.

Транспортный поток создается таким образом, чтобы с минимальными усилиями можно было бы выполнять над Транспортным потоком несколько операций. Среди них:

1) Получать кодированные данные одной из программ Транспортного потока, декодировать их и воспроизводить результаты декодирования, как показано рисунке Введение. 2.
2) Выделять пакеты Транспортного потока одной из программ Транспортного потока и формировать в качестве выходного сигнала другой Транспортный поток, содержащий только одну программу, как показано рисунке Введение. 3.
3) Выделять пакеты Транспортного потока одной или нескольких программ одного или нескольких Транспортных потоков и формировать в качестве выходного сигнала другой Транспортный поток (нет иллюстрации).
4) Выделять содержание одной из программ и формировать в качестве выходного сигнала Программный поток, содержащий эту программу, как показано рисунке Введение. 4.

5) Преобразовывать Программный поток в Транспортный поток для передачи его в условиях с большими потерями, и затем восстанавливать пригодный для использования, и, в некоторых случаях идентичный Программный поток.

На рисунке Ведение. 2 и рисунке Ведение. 3 показаны прототипы систем демультиплексирования и декодирования, на вход которых поступает Транспортный поток. На рисунке Ведение. 2 показан первый случай, в котором демультиплексируется и декодируется непосредственно Транспортный поток. Транспортные потоки состоят из двух уровней:

– системный уровень; и

– уровень компрессии.

Входной поток для декодера Транспортного потока содержит системный уровень, охватывающий уровень компрессии. Входные потоки для видео и аудио декодеров содержат только уровень компрессии.

Операции, выполняемые прототипом декодера, на вход которого поступают Транспортные потоки, относятся либо к Транспортному потоку в целом ("операции над всем мультиплексированным потоком"), либо к отдельным элементарным потокам ("операции над конкретным потоком"). Системный уровень Транспортного потока подразделяется на два подуровня, один – для операций над всем мультиплексированным потоком (Уровень пакетов Транспортного потока), и один – для операций над конкретным потоком (Уровень пакетов элементарного потока пакетов (PES)).

Прототип декодера для Транспортных потоков, включая аудио и видео потоки, также изображен на рисунке Введение. 2 и иллюстрирует работу декодера. Эта архитектура не является единственно возможной – некоторые функции системного декодера, такие как управление синхронизацией декодера, могут точно также быть распределены между декодерами элементарных потоков и канальным декодером, но этот рисунок очень удобен для обсуждения. Аналогичным образом, сообщение об ошибке, обнаруженной канальным декодером, может быть передано отдельным аудио и видео декодерам различными способами, и эти линии связи на диаграмме не показаны. Проект прототипа декодера не накладывает никаких нормативных требований на проект декодера Транспортного потока. Кроме того, допускаются также и данные, не являющиеся аудио/видео данными, хотя они на рисунке и не показаны.

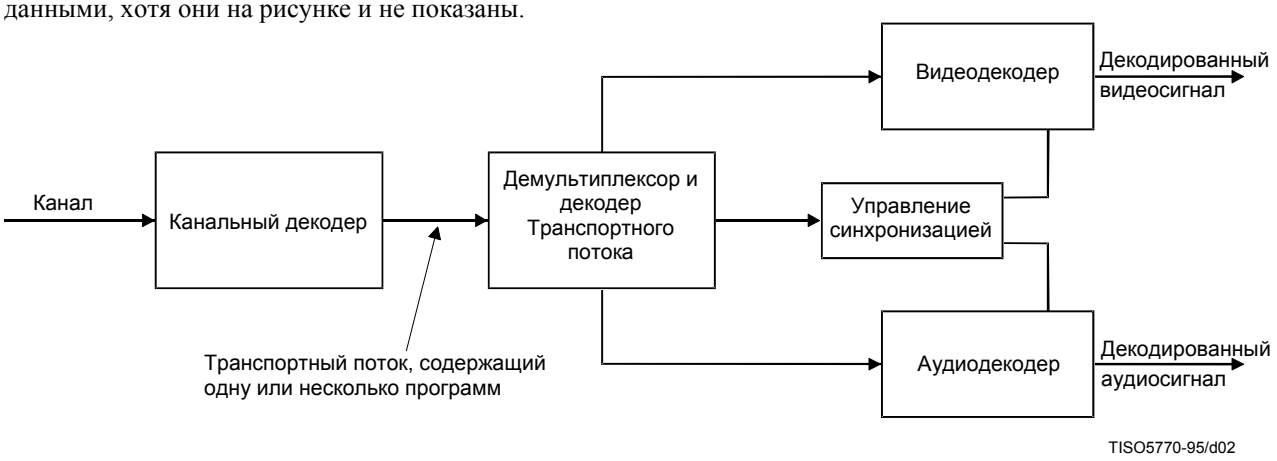


Рисунок Введение. 2 –
Пример прототипа демультиплексирования и декодирования транспортного потока

На рисунке Введение. 3 показан второй случай, в котором Транспортный поток, содержащий множество программ, преобразуется в Транспортный поток, содержащий одну программу. В таком случае операция повторного мультиплексирования может потребовать корректировки значений Эталонного времени программы (Program Clock Reference (PCR)), для того чтобы учесть изменение мест расположения полей PCR в двоичном потоке.

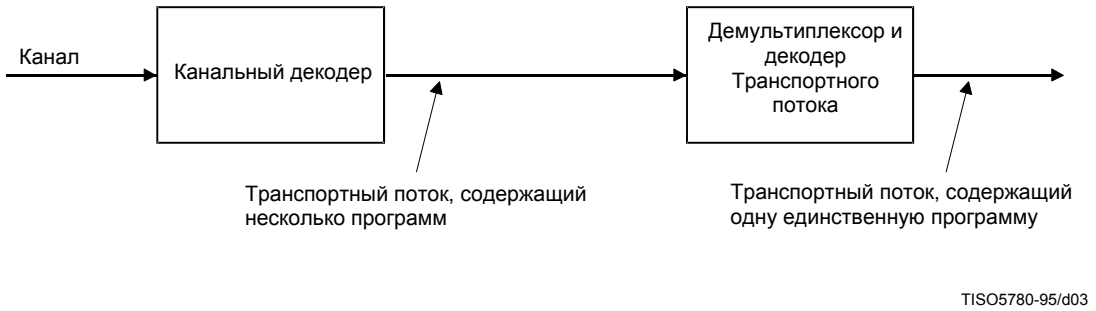


Рисунок Введение. 3 – Пример прототипа мультиплексирования транспортного потока

На рисунке Введение. 4 показан случай, в котором многопрограммный Транспортный поток сначала демультиплексируется, а затем преобразуется в Программный поток.

Рисунки Введение. 3 и Введение. 4 показывают, что вполне возможно и целесообразно выполнять преобразование между различными типами и конфигурациями Транспортных потоков. В синтаксисе **Транспортного потока** и **Программного потока** определены специальные поля, которые упрощают выполнение изображенных преобразований. Нет никаких требований относительно конкретных вариантов реализации демультиплексоров и декодеров, способных выполнять все эти функции.

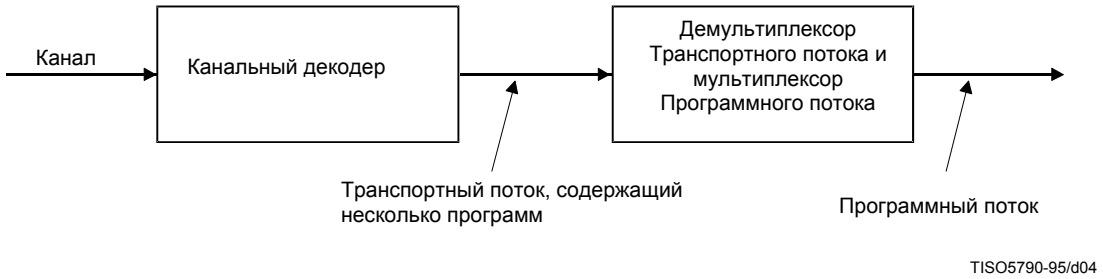


Рисунок Введение. 4 – Прототип преобразования Транспортного потока в Программный поток

Введение. 2 Программный поток

Программный поток – определяется, как поток, который предназначен для передачи или хранения одной программы кодированных данных и других данных в условиях, где вероятность возникновения ошибок чрезвычайно мала, и где основной задачей является обработка системных кодов, например, программным путем.

Программные потоки могут иметь либо фиксированную, либо переменную скорость передачи. В любом случае, составляющие их элементарные потоки могут иметь либо фиксированную, либо переменную скорость передачи. Ограничения потока по синтаксису и семантике для каждого из этих случаев одни и те же. Скорость передачи Программного потока определяется значениями и местами расположения полей Эталонного времени системы (System Clock Reference (SCR)) и полей mux_rate.

На рисунке Введение. 5 показан прототип системы аудио/видео декодера Программного потока. Эта архитектура не является единственно возможной – некоторые функции системного декодера, такие как управление синхронизацией декодера, могут точно также быть распределены между декодерами элементарных потоков и канальным декодером, но этот рисунок очень удобен для обсуждения. Проект прототипа декодера не накладывает никаких нормативных требований на проект декодера Программного потока. Кроме того, допускаются также и данные, не являющиеся аудио/видео данными, хотя они на рисунке и не показаны.

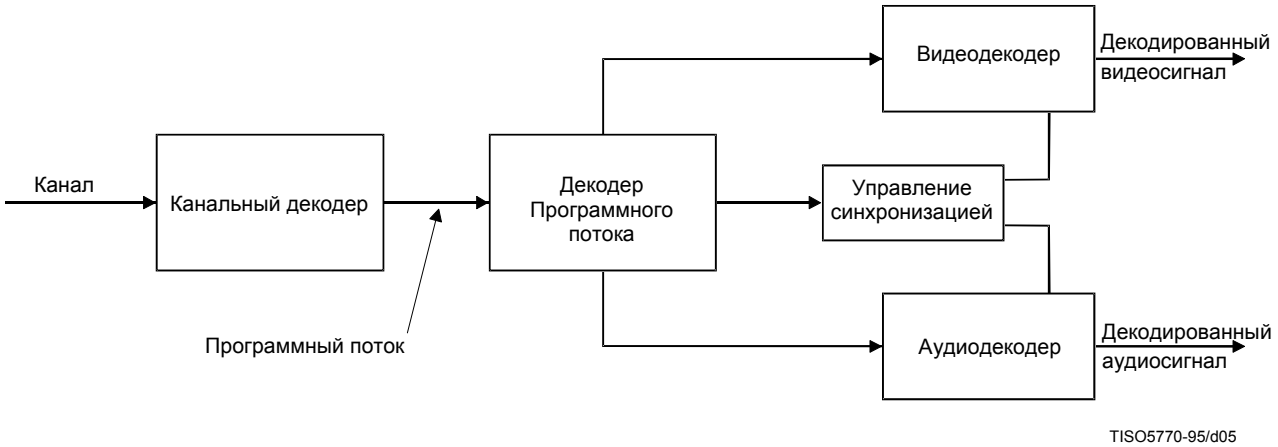


Рисунок Введение.5 – Прототип декодера для Программных потоков

Прототип декодера для Программных потоков, показанный на рисунке Введение. 5, состоит из системных видео и аудио декодеров, соответствующих Частям 1, 2 и 3 стандарта ИСО/МЭК 13818, соответственно. В этом декодере предполагается, что мультиплексированное кодированное воспроизведение одного или нескольких аудио и/или видео потоков сохраняется или передается по некоторому каналу в некотором формате, свойственном этому каналу. Этот свойственный каналу формат не определяется ни настоящей Рекомендацией | Международным стандартом, ни участком канального декодирования прототипа декодера.

Прототип декодера принимает на своем входе Программный поток и использует декодер Программного потока для выделения из потока информации синхронизации. Декодер Программного потока демультиплексирует поток, и полученные в результате этого элементарные потоки служат входными сигналами для видео и аудио декодеров, выходные сигналы которых представляют собой декодированные видео и аудио сигналы. В проекте учтена, хотя и не показана на рисунке, передача информации синхронизации между декодером Программного потока, видео и аудио декодерами и канальным декодером. Используя эту информацию синхронизации, видео и аудио декодеры синхронизируются друг с другом и с каналом передачи.

Программные потоки состоят из двух уровней: системного уровня и уровня компрессии. Входной поток декодера Программного потока содержит системный уровень, охватывающий уровень компрессии. Входные потоки для видео и аудио декодеров содержат только уровень компрессии.

Операции, выполняемые прототипом декодера, относятся либо к Программному потоку в целом ("операции над всем мультиплексированным потоком"), или к отдельным элементарным потокам ("операции над конкретным потоком"). Системный уровень Программного потока подразделяется на два подуровня, один – для операций над всем мультиплексированным потоком (Уровень пакетов), и один – для операций над конкретным потоком (Уровень пакетов PES).

Введение. 3 Преобразование между Транспортным потоком и Программным потоком

Может быть вполне возможным и целесообразным выполнять преобразование между Транспортными потоками и Программными потоками при помощи пакетов PES. Это вытекает из спецификации Транспортного потока и Программного потока в соответствии с положениями, прописанными в § 2.4.1 и § 2.5.1 нормативных требований настоящей Рекомендации | Международного стандарта. Пакеты PES, с некоторыми ограничениями, из полезной нагрузки одного мультиплексированного двоичного потока могут быть преобразованы непосредственно в полезную нагрузку другого мультиплексированного двоичного потока. Для того чтобы упростить эту задачу, можно идентифицировать правильный порядок следования пакетов PES в программе, если во всех пакетах PES содержится значение счетчика program_packet_sequence_counter.

Некоторая другая информация, необходимая для преобразования, например, соотношение между элементарными потоками, содержится в Таблицах и заголовках обоих потоков. Такие данные, если они доступны, должны быть корректными в любом потоке и до, и после преобразования.

Введение. 4 Элементарный поток пакетов

Транспортные потоки и Программные потоки являются логическими конструкциями, состоящими из пакетов PES, как указано в определениях синтаксиса в § 2.4.3.6. Для выполнения преобразования между Транспортными потоками и Программными потоками должны использоваться пакеты PES; в некоторых случаях при выполнении такого преобразования пакеты PES не требуют модификации. Пакеты PES могут быть намного больше пакетов Транспортного потока.

Для формирования потока PES может использоваться непрерывная последовательность пакетов PES одного элементарного потока с одним ID потока. Когда пакеты PES используются для формирования потока PES, они

должны содержать поля Эталонного времени элементарного потока (Elementary Stream Clock Reference (ESCR)) и поля скорости передачи элементарного потока (ES_Rate), ограничения для которых определены в § 2.4.3.8. Данные потока PES должны представлять собой смежные байты элементарного потока в их исходном порядке следования. Потоки PES не содержат некоторой необходимой системной информации, которая содержится в Программных и Транспортных потоках. Это, например, информация в заголовке пакета, системном заголовке, Таблице преобразования Программного потока, Каталоге Программного потока, Таблице преобразования программы и элементах синтаксиса пакетов Транспортного потока.

Поток PES – это логическая конструкция, которая может быть полезной в рамках вариантов реализации настоящей Рекомендации | Международного стандарта; однако, она не определяется как поток для обмена информацией и взаимодействия. Приложения, для которых требуются потоки, содержащие только один элементарный поток, могут использовать Программные потоки или Транспортные потоки, каждый из которых содержит только один элементарный поток. Эти потоки содержат всю необходимую системную информацию. Несколько Программных или Транспортных потоков, каждый из которых содержит один элементарный поток, могут быть сформированы с общей временной базой и, следовательно, они будут передавать полную программу, т. е. с видео и аудио данными.

Введение. 5 Модель синхронизации

Все видео и аудио системы имеют модель синхронизации, в которой сквозная задержка сигнала от входа кодера до выхода декодера является постоянной. Эта задержка представляет собой сумму задержек кодирования, буферизации в кодере, мультиплексирования, передачи или хранения, демультиплексирования, буферизации в декодере, декодирования и воспроизведения. В качестве одной из частей этой модели синхронизации все видеокадры и отсчеты аудиосигнала представлены только один раз, если только они не закодированы специальным образом так, чтобы этого не происходило, а межкадровый интервал и скорость передачи аудио отсчетов на декодере и на кодере являются одинаковыми. Системный код потока содержит информацию синхронизации, которая может использоваться для реализации систем, которые предусматривают постоянную сквозную задержку. Вполне возможно создать декодеры, которые не отвечают этой модели в точности; однако, в таких случаях ответственность за работу надлежащим образом лежит на декодере. Синхронизация включена в нормативные спецификации настоящей Рекомендации | Международного стандарта, которые должны выполняться для всех пригодных двоичных потоков, вне зависимости от способа их создания.

Все требования к синхронизации определены в условиях общего задающего генератора системы, который называется Системными часами. В Программном потоке частота этого генератора может иметь точно определенное соотношение с частотой синхронизации видеосигнала или отсчетов аудио сигнала, или может иметь рабочую частоту, которая слегка отличается от точного отношения, но все равно продолжает обеспечивать точное восстановление сквозной синхронизации и тактовой частоты.

В Транспортном потоке частота тактового генератора ограничена таким образом, что она всегда должна иметь точно определенное соотношение с частотой синхронизации видеосигнала или отсчетов аудиосигнала; результат этого ограничения заключается в упрощении восстановления частоты отсчетов в декодерах.

Введение. 6 Условный доступ

Шифрование и скремблирование для обеспечения условного доступа к программам, кодированным в виде Программных и Транспортных потоков, поддерживается определениями системного потока данных. Механизмы условного доступа здесь не определяются. Определения потока разработаны так, чтобы обеспечить разумные варианты реализации реальных систем условного доступа, а также имеется несколько синтаксических элементов, которые обеспечивают специальную поддержку для таких систем.

Введение. 7 Операции над всем мультиплексированным потоком

Операции над всем мультиплексированным потоком включают в себя координацию получения данных канала, настройку задающих генераторов и управление буферами. Эти задачи тесно связаны между собой. Если скорость доставки данных канала регулируется, то доставка данных может быть отрегулирована так, чтобы буферы декодера никогда бы не переполнялись и никогда бы не были недозаполненными; но если скорость доставки данных канала не регулируется, то декодеры элементарных потоков во избежание переполнения или недозаполнения, синхронизироваться данными, принимаемыми из канала.

Программные потоки состоят из пакетов, заголовки которых упрощают выполнение вышеописанных задач, заголовки пакетов указывают необходимые моменты времени, когда каждый байт должен войти в декодер Программного потока из канала, и это расписание прибытия служит справочным руководством для настройки задающих генераторов и управления буферами. Это расписание не должно выполняться декодерами в точности, но они должны компенсировать все отклонения от него.

Аналогично, Транспортные потоки состоят из пакетов Транспортного потока, заголовки которых содержат информацию, которая определяет моменты времени, когда каждый байт должен войти из канала в декодер Транспортного потока. Это расписание выполняет точно такие же функции, что были определены в Программном потоке.

Дополнительной операцией, выполняемой над всем мультиплексированным потоком, является способность декодера определить, какие ресурсы необходимы для декодирования Транспортного потока или Программного потока. Первый пакет каждого Программного потока содержит параметры, которые помогают декодерам выполнить эту задачу. Например, это максимальная скорость передачи данных в потоке или наибольшее число одновременно передаваемых видеоканалов. Кроме того, Транспортный поток содержит информацию, полезную на глобальном уровне.

Каждый Транспортный и Программный поток содержит информацию, которая определяет соответствующие характеристики элементарных потоков и взаимосвязь между элементарными потоками, образующими каждую программу. Такая информация может содержать данные о языке, на котором ведется передача в аудиоканалах, а также о взаимосвязях между видеопотоками, кода используется многоуровневое видеокодирование.

Введение. 8 Операции над конкретным потоком (Уровень пакетов PES)

Принципиально важными операциями над конкретным потоком являются:

1) демультиплексирование; и
2) синхронизированное воспроизведение множества элементарных потоков.

Введение. 8.1 Демультиплексирование

В процессе кодирования Программные потоки формируются путем мультиплексирования элементарных потоков, и Транспортные потоки формируются путем мультиплексирования элементарных потоков, Программных потоков, или элементов содержания других Транспортных потоков. Элементарные потоки, в дополнение к аудио и видео потокам, могут содержать конфиденциальные и зарезервированные потоки, а также потоки заполнения. Во временной области эти потоки делятся на пакеты, и пакеты передаются последовательно. Пакет PES содержит кодированные байты из одного и только одного элементарного потока.

В Программном потоке допускается наличие пакетов как фиксированной, так и переменной длины, в зависимости от ограничений, определенных в § 2.5.1 и § 2.5.2. Для Транспортных потоков длина пакета равна 188 байтов. Допускается наличие пакетов PES как фиксированной, так и переменной длины, и в большинстве приложений они будут относительно длинными.

В процессе декодирования, требуется выполнить демультиплексирование, в результате чего восстановить элементарные потоки из мультиплексированного Программного потока или Транспортного потока. Это возможно за счет использования кодов Stream_id в заголовках пакетов Программного потока и кодов ID пакетов в Транспортном потоке.

Введение. 8.2 Синхронизация

Синхронизация между множеством элементарных потоков осуществляется при помощи меток времени воспроизведения (Presentation Time Stamps (PTS)) в Программных и Транспортных потоках. Метки времени, обычно, выражаются в единицах по 90 кГц, но эталонное время системы (SCR), эталонное время программы (PCR) и дополнительное эталонное время элементарного потока (ESCR) имеют расширенные представления с разрешением 27 МГц. Декодирование N-элементных потоков синхронизируется путем настойки процесса декодирования потоков в соответствии с общей ведущей временной базой, а не путем настойки декодирования одного потока так, чтобы оно соответствовало декодированию другого потока. Ведущей временной базой может быть один из задающих генераторов N декодеров, задающий генератор источника данных, или некоторый внешний задающий генератор.

Каждая программа в Транспортном потоке, который может содержать множество программ, может иметь свою собственную временную базу. Временные базы различных программ в Транспортном потоке могут быть различными.

Поскольку при декодировании отдельных элементарных потоков используются метки времени воспроизведения (PTS), они размещаются на уровне пакетов PES как Транспортных, так и Программных потоков. Сквозная синхронизация осуществляется, когда во время получения информации кодеры сохраняют метки времени, когда метки времени передаются на декодеры вместе с соответствующими кодированными данными, и когда декодеры используют эти метки времени для составления расписания представления информации.

Синхронизация системы декодирования при помощи канала выполняется путем использования в Программном потоке Эталонного времени системы (SCR), а в Транспортном потоке – его аналога – Эталонного времени программы (PCR). SCR и PCR представляют собой метки времени, кодирующие синхронизацию самого двоичного потока, они получаются из той же временной базы, которая используется в этой программе для получения значений PTS для видео и аудио сигналов. Поскольку каждая программа может иметь свою собственную временную базу, в Транспортном потоке, содержащем множество программ, для каждой программы предусмотрены раздельные поля PCR. В некоторых случаях может допускаться, чтобы программы использовали поля PCR совместно. Метод идентификации того, какой PCR связан с данной программой,

описан в § 2.4.4 "Специальная информация программы (PSI)". Программа должна иметь одну и только одну временную базу PCR, которая с ней связана.

Введение. 8.3 Связь с уровнем компрессии

Уровень пакетов PES в некотором смысле независим от уровня компрессии, но не совсем. Он независим в том смысле, что полезная нагрузка пакета PES не должна начинаться в момент начала стартовых кодов уровня компрессии, которые определены в Частях 2 и 3 стандарта ИСО/МЭК 13818. Например, стартовые коды видеосигнала могут появляться в любом месте полезной нагрузки пакета PES, и стартовые коды могут быть разделены заголовком пакета PES. Однако метки времени, закодированные в заголовках пакетов PES, применяются к моментам времени воспроизведения конструкций уровня компрессии (а именно, модулям представления). Кроме того, когда данные элементарного потока соответствуют Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 13818-3, поле PES_packet_data_bytes должно быть побайтно синхронизировано с байтами настоящей Рекомендации | Международного стандарта.

Введение. 9 Эталонный системный декодер

В Части 1 стандарта ИСО/МЭК 13818 для формализации взаимосвязей по синхронизации и буферизации используется понятие "Декодер конечной системы" (System Target Decoder (STD)), один – для Транспортных потоков (см. § 2.4.2), который называется "Транспортный декодер конечной системы" (Transport System Target Decoder (T-STD)) и один – для Программных потоков (см. § 2.5.2), который называется "Программный декодер конечной системы" (Program System Target Decoder (P-STD)). Поскольку параметры декодера STD (например, размеры буферов) определены в полях, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, каждый элементарный поток осуществляет собственную параметризацию декодера STD. Кодеры должны создавать двоичные потоки, который отвечают соответствующим ограничениям, наложенным декодером STD. Физические декодеры могут предполагать, что поток правильно воспроизводится на его STD. Физический декодер должен обеспечивать компенсацию расхождений своего состава с составом декодера STD.

Введение. 10 Приложения

Потоки, определенные в настоящей Рекомендации | Международном стандарте, предназначены для того, чтобы быть максимально возможно полезными для широкого спектра приложений. Разработчикам приложений следует выбирать наиболее пригодный поток.

Современные сети передачи данных могут поддерживать передачу видеопотоков, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, и передачу аудиопотоков, соответствующих с ИСО/МЭК 13818. Требуется протокол транспортировки в реальном времени. Для передачи данных в таких сетях может быть пригоден Программный поток.

Программный поток пригоден также для мультимедийных приложений на CD-ROM. Может применяться программная обработка Программного потока.

Транспортный поток может быть более пригодным для условий передачи, подверженных ошибкам, например, в сетях передачи компрессированных видеопотоков на большие расстояния, и в радиовещательных системах.

Для многих приложений требуется хранение данных в различных цифровых запоминающих устройствах (Digital Storage Media (DSM)) и извлечение из памяти двоичных потоков, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Протокол управления и контроля цифровых запоминающих устройств (DSM-CC) определяется в Приложении B и Части 6 стандарта ИСО/МЭК 13818 и имеет своей целью упрощение управления этими устройствами.

МЕЖДУНАРОДНЫЙ СТАНДАРТ
РЕКОМЕНДАЦИЯ МСЭ-Т

Информационная технология – Общее кодирование подвижных изображений и соответствующей аудиоинформации: Системы

РАЗДЕЛ 1 – ОБЩИЕ ПОЛОЖЕНИЯ

1.1 Сфера применения

В настоящей Рекомендации | Международном стандарте определяется спецификация системного уровня кодирования. Она была разработана, главным образом, в целях обеспечения поддержки объединения методов синхронизации кодирования изображения и звука, определенных в Частях 2 и 3 стандарта ИСО/МЭК 13818. Системный уровень поддерживает шесть основных функций:

1) синхронизация множества компрессированных потоков при декодировании;
2) перемежение множества компрессированных потоков в единый поток;
3) запуск процесса буферизации, для того чтобы начать декодирование;
4) непрерывное управление буфером;
5) идентификация времени; и
6) мультиплексирование и передача сигналов различных компонентов системного потока.

Мультиплексированный двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, представляет собой либо **Транспортный поток**, либо **Программный поток**. Оба потока состоят из **пакетов PES** и пакетов, содержащих другую необходимую информацию. Оба типа потоков поддерживают мультиплексирование компрессированных потоков изображения и звука одной программы с общей временной базой. **Транспортный поток** дополнительно поддерживает мультиплексирование компрессированных потоков изображения и звука нескольких программ с независимыми временными базами. **Программный поток,** как правило, более приемлем для условий работы "почти без ошибок", так как он обеспечивает возможность компьютерной обработки программной информации. Транспортный поток более пригоден для использования в условиях, где вероятно появление помех.

Мультиплексированный двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, вне зависимости от того, является ли он Транспортным или Программным, состоит из двух уровней: внешним уровнем является системный уровень, а внутренним – уровень компрессии. Системный уровень обеспечивает выполнение функций, необходимых для использования в системе одного или нескольких компрессированных потоков данных. Части настоящей спецификации, касающиеся изображения и звука, определяют уровень компрессии кодирования для данных изображения и звука. Кодирование других типов данных в настоящей Рекомендации | Международном стандарте не определяется, но поддерживается системным уровнем, при условии, что другие типы данных соответствуют ограничениям, определенным § 2.7.

1.2 Нормативные справочные документы

В нижеследующих Рекомендациях МСЭ-Т и других справочных документах содержатся положения, которые, посредством ссылок в настоящем тексте, составляют положения настоящей Рекомендации | Международного стандарта. На время публикации указанные здесь издания были действительными. Все Рекомендации и Стандарты постоянно пересматриваются, и сторонам – участницам соглашений, основанных на настоящей Рекомендации | Международном стандарте, настоятельно рекомендуется изучить возможность использования последних изданий перечисленных ниже Рекомендаций и Стандартов. Члены МЭК и ИСО поддерживают регистры Международных стандартов, действующих на настоящий момент. Бюро стандартизации электросвязи МСЭ поддерживает перечень действующих на настоящий момент Рекомендаций МСЭ-Т.

1.2.1 Идентичные Рекомендации | Международные стандарты

– Рекомендация МСЭ-Т H.262 (2000 г.) | ИСО/МЭК 13818-2:2000, *Информационная технология – Общее кодирование подвижных изображений и соответствующей аудиоинформации: Видео*.

1.2.2 Парные Рекомендации | Международные стандарты с равнозначным техническим содержанием

– Рекомендация МСЭ-Т H.264 (2005 г.), *Улучшенное кодирование видеосигнала для основополагающих аудиовизуальных услуг*.

  ИСО/МЭК 14496-10:2005, *Информационная технология – Кодирование аудиовизуальных объектов – Часть 10: Улучшенное кодирование видеосигнала*.

– Рекомендация МСЭ-Т Т.171 (1996 г.), *Протоколы для интерактивных аудиовизуальных: кодированное представление мультимедийных и гипермедийных объектов*.

ИСО/МЭК 13522-1:1997, *Информационная технология – Кодирование мультимедийной и гипермедийной информации – Часть 1: Представление объектов MHEG – базовая нотация (ASN.1)*.

1.2.3 Дополнительные справочные документы

– ИСО 639-2:1998, *Коды для представления названий языков – Часть 2: Код Альфа-3*.

– ИСО 8859-1:1998, *Информационная технология – 8-битовые однобайтовые наборы графических символов – Часть 1: Латинской алфавит №. 1*.

– ИСО 15706:2002, *Информация и документация – Международный стандартный номер аудиовизуального произведения (ISAN)*.

– ИСО/PRF 15706-2, *Информация и документация – Международный стандартный номер аудиовизуального произведения (ISAN) – Часть 2: Идентификатор версии*.

– ИСО/МЭК 11172-1:1993, *Информационная технология – Кодирование подвижных изображений и соответствующей аудиоинформации для цифровых сред хранения со скоростями до примерно 1,5 Мбит/с– Часть 1: Системы*.

– ИСО/МЭК 11172-2:1993, *Информационная технология – Кодирование подвижных изображений и соответствующей аудиоинформации для цифровых сред хранения со скоростями до примерно 1,5 Мбит/с – Часть 2: Видео*.

– ИСО/МЭК 11172-3:1993, *Информационная технология – Кодирование подвижных изображений и соответствующей аудиоинформации для цифровых сред хранения со скоростями до примерно 1,5 Мбит/с– Часть 3: Аудио*.

– ИСО/МЭК 13818-3:1998, *Информационная технология – Общее кодирование подвижных изображений и соответствующей аудиоинформации – Часть 3: Аудио*.

– ИСО/МЭК 13818-6:1998, *Информационная технология – Общее кодирование подвижных изображений и соответствующей аудиоинформации – Часть 6: Расширения для DSM-CC*.

– ИСО/МЭК 13818-7:2006, *Информационная технология – Общее кодирование подвижных изображений и соответствующей аудиоинформации – Часть 7: Улучшенное кодирование аудиосигнала (AAC)*.

– ИСО/МЭК 13818-11:2004, *Информационная технология – Общее кодирование подвижных изображений и соответствующей аудиоинформации – Часть 11: IPMP в системах MPEG-2*.

– ИСО/МЭК 14496-1:2004, *Информационная технология – Кодирование аудиовизуальных объектов – Часть 1: Системы*.

– ИСО/МЭК 14496-2:2004, *Информационная технология – Кодирование аудиовизуальных объектов – Часть 2: Визуальные объекты*.

– ИСО/МЭК 14496-3:2005, *Информационная технология – Кодирование аудиовизуальных объектов – Часть 3: Аудио*.

– Рекомендация МСЭ-R BT.601-6 (2007 г.), *Параметры студийного кодирования цифрового телевидения для стандартного 4:3 и широкоэкранного 16:9 растров.*

– Рекомендация МСЭ-R BT.470-7 (2005 г.), *Традиционные системы аналогового телевидения*.

– Рекомендация МСЭ-R BR.648, *Цифровая запись звуковых сигналов*.

– Рекомендация МСЭ-Т J.17 (1988 г.), *Предыскажения, используемые в линиях передачи звуковых программ*.

– Публикация МЭК 60908:1999, *Звукозапись – Цифровая аудиосистема с компакт-дисками*.

РАЗДЕЛ 2 – ТЕХНИЧЕСКИЕ ЭЛЕМЕНТЫ

2.1 Определения

В настоящей Рекомендации | Международном стандарте применяются следующие определения. Если определение относится к определенной Части, это указывается в скобках.

2.1.1 access unit (system) AVC – модуль доступа AVC (система): Кодированное представление модуля представления. Для аудиосигнала модулем доступа является кодированное представление звукового кадра.

Для видеосигнала модуль доступа включает в себя все кодированные данные некоторого кадра и все данные заполнения, которые следуют за ним, но не включает начало следующего модуля доступа. Если кадру не

предшествует код group_start_code или код sequence_header_code, то модуль доступа начинается со стартового кода кадра. Если кадру предшествует код group_start_code и/или код sequence_header_code, то модуль доступа начинается с первого байта первого из этих стартовых кодов. Если это – последний кадр, которому в двоичном потоке предшествует код sequence_end_code, то все байты между последним байтом кодированного кадра и кодом sequence_end_code (включая код sequence_end_code) принадлежат модулю доступа.

В Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 "Видео" модуль доступа AVC определен в п. 2.1.3.

2.1.2 AVC 24-hour picture (system) – AVC с 24-часовым изображением (система): В будущем – модуль доступа AVC с временем воспроизведения более 24 часов. Согласно этому определению, модуль доступа AVC в будущем будет иметь время воспроизведения более 24 часов, если разница между начальным временем прихода $t_{ai}(n)$ и временем выхода DPB $t_{o,dpb}(n)$ больше 24 часов.

2.1.3 AVC access unit (system) – модуль доступа AVC (система): Модуль доступа, как он определен для потоков байтов в Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 с ограничениями, изложенными в § 2.14.1.

2.1.4 AVC Slice (system) – секция (система) AVC: Модуль byte_stream_nal_unit, определенный в Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, со значением параметра nal_unit_type, равным 1 или 5, или структура данных byte_stream_nal_unit со значением параметра nal_unit_type, равным 2, и любая связанная с ними структура данных byte_stream_nal_unit со значением параметра nal_unit_type, равным 3 и/или 4.

2.1.5 AVC still picture (system) – неподвижное изображение (система) AVC: неподвижное изображение AVC состоит из модуля доступа AVC, содержащего изображение IDR, которому предшествуют модули SPS и PPS NAL, в которых передается информация, достаточная для корректного декодирования изображения IDR. Перед неподвижным изображением AVC должно быть другое неподвижное изображение AVC или модуль NAL "Конец последовательности", которым заканчивается предыдущая закодированная последовательность, если только неподвижное изображение AVC не является самым первым блоком видеопотока.

2.1.6 AVC video sequence (system) – видеопоследовательность (система) AVC: Кодированная видеопоследовательность, определенная в Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 п. 3.27.

2.1.7 AVC video stream (system) – видеопоток AVC (система): Поток, соответствующий Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Видеопоток AVC состоит из одной или нескольких видеопоследовательностей.

2.1.8 bitrate – скорость передачи: Скорость, с которой компрессированный двоичный поток доставляется из канала на вход декодера.

2.1.9 byte aligned – выровненный побайтно: Бит в кодированном двоичном потоке является побайтно выровненным, если его положение относительно первого бита потока кратно 8 битам.

2.1.10 channel – канал: Цифровая среда, в которой сохраняется или транспортируется поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

2.1.11 coded B-frame – кодированный кадр B: Изображение кадра B или пара изображений поля B.

2.1.12 coded frame – кодированный кадр: Кодированный кадр – это кодированный кадр I, или кодированный кадр B, или кодированный кадр P.

2.1.13 coded I-frame – кодированный кадр I: Изображение кадра I или пара полей изображения, в которой первое поле является изображением I, а второе поле – изображением I или изображением P.

2.1.14 coded P-frame – кодированный кадр P: Изображение кадра P или пара полей изображения P.

2.1.15 coded representation – кодированное представление: Элемент данных, представленный в своей кодированной форме.

2.1.16 compression – компрессия: Уменьшение числа битов, используемых для представления объекта данных.

2.1.17 constant bitrate – постоянная скорость передачи: Операция, при которой скорость передачи постоянна он начала до окончания компрессированного двоичного потока.

2.1.18 constrained system parameter stream; CSPS (system) – поток с ограниченными системными параметрами (система): Программный поток, к которому применяются ограничения, определенные в § 2.7.9.

2.1.19 Cyclic Redundancy Check (CRC) – Циклическая проверка по избыточности: Циклическая проверка по избыточности для проверки корректности данных.

2.1.20 data element – элемент данных: Объект данных, в своем исходном представлении до кодирования или после декодирования.

2.1.21 decoded stream – декодированный поток: Восстановленная после декодирования версия компрессированного двоичного потока.

2.1.22 decoder – декодер: Материальная реализация процесса декодирования.

2.1.23 decoding (process) – декодирование (процесс): Процесс, определенный в настоящей Рекомендации | Международном стандарте, на вход которого поступает кодированный двоичный поток и который создает на своем выходе декодированные изображения или отсчеты аудиосигнала.

2.1.24 decoding time-stamp; DTS (system) – метка времени декодирования (система): Поле, которое может присутствовать в заголовке пакета PES, и которое указывает момент времени, когда модуль доступа декодируется в декодере конечной системы.

2.1.25 digital storage media (DSM) – цифровое запоминающее устройство: Цифровое устройство или система для хранения или передачи данных.

2.1.26 DSM-CC – Контроль и управление цифрового запоминающего устройства: Контроль и управление цифрового запоминающего устройства.

2.1.27 entitlement control message (ECM) – Сообщение, управляющее правами доступа: Сообщения, управляющие правами – это конфиденциальная информация условного доступа, которая определяет управляющие слова и, возможно, другие, как правило, определяемые потоком, параметры скремблирования и/или управления.

2.1.28 entitlement management message (EMM) – Сообщение, предоставляющее право доступа: Сообщения, предоставляющие право доступа – это конфиденциальная информация условного доступа, которая определяет уровни авторизации или услуги определенных декодеров. Они могут быть адресованы отдельным декодерам или группам декодеров.

2.1.29 editing – монтаж: Процесс, посредством которого один или несколько компрессированных двоичных потоков видоизменяются в целях создания нового компрессированного двоичного потока. Смонтированные двоичные потоки соответствуют тем же требованиям, что и не смонтированные двоичные потоки.

2.1.30 elementary stream; ES (system) – элементарный поток (система): Общий термин для обозначения в пакетах PES одного из кодированных потоков – видео, аудио или другого кодированного двоичного потока. Один элементарный поток передается в виде последовательности пакетов PES с одним и только одним идентификатором потока stream_id.

2.1.31 Elementary Stream Clock Reference; ESCR (system) – эталонное время элементарного потока (система): Метка времени в потоке пакетов PES, из которой декодеры потока пакетов PES могут получить данные о синхронизации.

2.1.32 encoder – кодер: Материальная реализация процесса кодирования.

2.1.33 encoding (process) – кодирование (процесс): Процесс, не определенный в настоящей Рекомендации | Международном стандарте, на вход которого поступают изображения или отсчеты аудиосигнала и который создает на своем выходе кодированный двоичный поток, соответствующий настоящей Рекомендации.

2.1.34 entropy coding – энтропийное кодирование: Кодирование цифрового представления сигнала с переменной длиной и без потерь для уменьшения избыточности.

2.1.35 event – событие: Событие определяется как набор элементарных потоков с общей временной базой, соответствующего времени начала и времени окончания.

2.1.36 fast forward playback (video) – ускоренное воспроизведение (видесигнала): Процесс воспроизведения последовательности кадров, при котором скорость воспроизведения превышает скорость в реальном времени.

2.1.37 forbidden – запрещено: Термин "запрещено", когда он используется в разделах настоящей Рекомендации | Международного стандарта, определяющих кодированный двоичный поток, указывает, что обозначенное значение никогда не должно использоваться.

2.1.38 metadata: метаданные Информация, описывающая аудиовизуальный контент и основные данные в формате, определенном ИСО или другой уполномоченной организацией.

2.1.39 metadata access unit – модуль доступа метаданных: Глобальная структура внутри метаданных, которая определяет ту часть метаданных, которая должна быть декодирована в определенный момент времени. Внутренняя структура модуля доступа метаданных определяется форматом метаданных.

2.1.40 metadata application format – формат приложения метаданных: Определяет формат приложения, в котором используются метаданные; указывает информацию, присущую приложению, для транспортировки метаданных.

2.1.41 metadata decoder configuration information – информация о конфигурации декодера метаданных: Данные, необходимые приемнику для декодирования конкретной услуги метаданных. В зависимости от формата метаданных, информация о конфигурации декодера может требоваться или не требоваться.

2.1.42 metadata format – формат метаданных: Определяется формат кодирования метаданных.

2.1.43 metadata service – услуга метаданных: Когерентный комплект метаданных одинакового формата, доставленных на приемник с определенной целью.

2.1.44 metadata service id – идентификатор услуги метаданных: Идентификатор определенной услуги метаданных; используется в некоторых способах транспортировки метаданных.

2.1.45 metadata stream – поток метаданных: Объединение или набор модулей доступа метаданных одной или нескольких услуг метаданных.

2.1.46 (multiplexed) stream (system) – (мультиплексированный) поток (система): Двоичный поток, состоящий из 0 или нескольких элементарных потоков, объединенных таким способом, который соответствует настоящей Рекомендации | Международному стандарту.

2.1.47 layer (video and systems) – уровень (видео и системы): Один из уровней в иерархии данных спецификаций видео или систем, определенных в Частях 1 и 2 настоящей Рекомендации | Международного стандарта.

2.1.48 pack (system) – блок (система): Блок состоит из заголовка пакета, после которого следует ноль или несколько пакетов. Он представляет собой один из уровней синтаксиса системного кодирования, описанного в § 2.5.3.3.

2.1.49 packet data (system) – данные пакета (система): Непрерывная последовательность байтов данных элементарного потока, представленного в пакете.

2.1.50 packet identifier; PID (system) – идентификатор пакета (система): Уникальное целочисленное значение, используемое для идентификации элементарных потоков программы в однопрограммном или многопрограммном Транспортном потоке, как описано в § 2.4.3.

2.1.51 padding (audio) – заполнение (аудио): Метод регулировки средней по времени длины аудиокадра в соответствии с продолжительностью соответствующих ИКМ отсчетов, заключающийся в добавлении в аудиокадр – при выполнении некоторого условия – дополнительного временного интервала.

2.1.52 payload – полезная нагрузка: Полезной нагрузкой называются байты, которые расположены в пакете после байтов заголовка. Например, полезная нагрузка некоторых пакетов Транспортного потока содержит заголовок PES_packet_header и байты PES_packet_data_bytes, или поле pointer_field и разделы PSI, или конфиденциальную информацию; но полезная нагрузка PES_packet_payload состоит только из байтов данных PES_packet_data_bytes. Заголовок пакета Транспортного потока и поля адаптации полезной нагрузкой не являются.

2.1.53 PES (system) –PES (система): Сокращение термина "Packetized Elementary Stream" – элементарный поток пакетов.

2.1.54 PES packet (system) – пакет PES (система): Структура данных, используемая для передачи данных элементарного потока. Пакет PES состоит из заголовка пакета PES, после которого следует множество непрерывных байтов элементарного потока данных. Он представляет собой один из уровней синтаксиса системного кодирования, описанного в § 2.4.3.6.

2.1.55 PES packet header (system) – заголовок пакета PES (система): Первые поля пакета PES до полей PES_packet_data_byte, которые в заголовок не включаются, если поток не является потоком с заполнением. В случае потока с заполнением, заголовок пакета PES аналогичным образом определяется как первые поля пакета до полей padding_byte, которые в заголовок не включаются.

2.1.56 PES stream (system) – поток PES (система): Поток PES состоит из пакетов PES, полезная нагрузка которых состоит из данных одного-единственного элементарного потока, и имеющих один и тот же идентификатор потока stream_id. Применяются определенные семантические ограничения. См. Введение. 4.

2.1.57 presentation time-stamp; PTS (system) – метка времени представления (воспроизведения); (система) PTS: Поле, которое может присутствовать в заголовке пакета PES, и которое указывает время, когда модуль представления появляется в конечном декодере системы.

2.1.58 presentation unit; PU (system) – модуль представления; (система): Декодированный аудио модуль доступа или декодированный кадр изображения.

2.1.59 program (system) – программа (система): Программа – это набор элементов программы. Элементами программы могут быть элементарные потоки. Элементы программы не должны иметь какой-либо определенной временной базы; те, которые ее имеют, имеют общую временную базу и предназначены для синхронного воспроизведения.

2.1.60 Program Clock Reference; PCR (system) – Эталонное время программы (система): Метка времени в Транспортном потоке, на основании которой декодер получает информацию синхронизации.

2.1.61 program element (system) – элемент программы (система): Общий термин для одного из элементарных потоков или других потоков данных, которые могут быть включены в программу.

2.1.62 Program Specific Information; PSI (system) – Специальная информация программы; (система) PSI: PSI состоит из нормативных данных, которые необходимы для демультиплексирования Транспортных потоков и успешного восстановления программ, она описана в § 2.4.4. Примером неофициально определенной информации PSI является необязательная таблица сетевой информации.

2.1.63 random access – случайный доступ: Процесс начала чтения и декодирования кодированного двоичного потока в произвольной точке.

2.1.64 reserved – зарезервировано: Термин "зарезервировано", когда он используется в разделах, определяющих кодированный двоичный поток, указывает, что это значение может использоваться в будущем для расширений, определяемых ИСО. Если в тексте настоящей Рекомендации | Международного стандарта не определено иного, все зарезервированные биты должны быть установлены в '1'.

2.1.65 scrambling (system) – скремблирование (система): Изменение характеристик аудиопотока, видеопотока или потока кодированных данных с целью предотвращения несанкционированного приема информации в открытом виде. Это изменение представляет собой определенный в спецификации процесс, выполняемый под управлением системы условного доступа.

2.1.66 source stream – поток источника: Отдельный немультиплексированный поток отсчетов до компрессионного кодирования.

2.1.67 splicing (system) – сращивание (система): Объединение двух различных элементарных потоков, выполненное на системном уровне. Результирующий системный поток полностью соответствует настоящей Рекомендации | Международному стандарту. Монтаж может приводить к нарушению непрерывности временной базы, непрерывности счетчика, специальной информации программы (PSI) и декодирования.

2.1.68 start codes (system) – стартовые коды (система): 32-битовые коды, встроенные в кодированный двоичный поток. они используются для различных целей, включая идентификацию некоторых уровней в синтаксисе кодирования. Стартовые коды состоят из 24-битового префикса (0x000001) и одного 8-битового идентификатора потока stream_id, как показано в Таблице 2-22.

2.1.69 STD input buffer (system) – входной буфер конечного декодера (система): буфер "первым прибыл – первым обслужен" на входе декодера конечной системы для хранения компрессированных данных элементарных потоков до декодирования.

2.1.70 still picture – неподвижное изображение: Закодированное неподвижное изображение состоит из видеопоследовательности, закодированной, как определено в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, ИСО/МЭК 11172-2 или ИСО/МЭК 14496-2, которая содержит ровно одно закодированное изображение во внутренней кодировке. Это изображение имеет связанный с ним PTS, и в случае его кодирования в соответствии с ИСО/МЭК 11172-2, Рекомендацией МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 14496-2, время воспроизведения последовательных изображений, если таковые есть, отстает от времени воспроизведения неподвижного изображения, по крайней мере, на два периода кадровой развёртки.

2.1.71 system header (system) – системный заголовок (система): Системный заголовок – это структура данных, определенная в § 2.5.3.5, в которой передается информация, описывающая системные характеристики Программного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

2.1.72 System Clock Reference; SCR (system) – Эталонное время системы (система): Метка времени в Программном потоке, на основании которой декодер получает информацию синхронизации.

2.1.73 system target decoder; STD (system) – конечный декодер системы (система): Гипотетическая эталонная реализация процесса декодирования, используемая для определения семантики мультиплексированного двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

2.1.74 time-stamp (system) – метка времени (система): Термин, который указывает момент времени выполнения конкретной операции, например, прибытия байта или воспроизведения модуля представления.

2.1.75 Transport stream packet header (system) – Заголовок пакета Транспортного потока (система): Начальные поля в пакете Транспортного потока, до поля continuity_counter включительно.

2.1.76 variable bitrate – переменная скорость передачи: Атрибут Транспортных потоков или Программных потоков, в которых скорость прибытия байтов на вход декодера меняется во времени.

2.2 Символы и сокращения

Математические операторы, используемые для описания настоящей Рекомендации | Международного стандарта, аналогичны тем, что используются в языке программирования C. Однако, целочисленное деление с отсечением дробной части и округлением здесь определены иначе. Операции над битами определены в предположении, что целые числа представлены поразрядно дополнением до двух. Нумерация и подсчеты, как правило, начинаются с 0.

2.2.1 Арифметические операторы

+	Сложение
–	Вычитание (двоичный оператор) или отрицание (унарный оператор)
++	Пошаговое приращение
– –	Пошаговое уменьшение

* или ×	Умножение
^	Возведение в степень
/	Целочисленное деление с усечением результата в сторону нуля. Например, 7/4 и –7/–4 усекаются до 1, а –7/4 и 7/–4 усекаются до –1.
//	Целочисленное деление с округлением до ближайшего целого числа. Половины целочисленных значений округляются в сторону от нуля, если не указано иное. Например, 3//2 округляется до 2, а –3//2 округляется до –2.
DIV	Целочисленное деление с усечением результата в сторону $-\infty$.
%	Оператор модуля. Определен только для положительных чисел.
Sign( )	Sign(x) = 1 x > 0 0 x = = 0 –1 x < 0
NINT( )	Оператор "ближайшее целое число". Возвращает ближайшее целочисленное значение вещественного аргумента. Половины целочисленных значений округляются в сторону от нуля.
sin	Синус
cos	Косинус
exp	Экспонента
√	Корень квадратный
$\log_{10}$	Логарифм по основанию 10
$\log_e$	Логарифм по основанию e

2.2.2 Логические операторы

||	Логическое ИЛИ
&&	Логическое И
!	Логическое НЕ

2.2.3 Операторы сравнения

>	Больше
≥	Больше или равно
<	Меньше
≤	Меньше или равно
= =	Равно
!=	Не равно
max [,...,]	Максимальное значение из списка аргументов
min [,...,]	Минимальное значение из списка аргументов

2.2.4 Побитовые операторы

&	И
|	ИЛИ
>>	Сдвиг вправо с введением дополнительного знакового разряда
<<	Сдвиг влево с заполнением нулями

2.2.5 Присвоение значений

=	Оператор присвоения

2.2.6 Мнемокоды

Нижеприведенные мнемокоды определены для описания различных типов данных, использованных в кодированном двоичном потоке.

bslbf	Строка битов, в которой левый бит является первым, где "левый" означает порядок, в котором строки битов записаны в настоящей Рекомендации | Международном стандарте. Строки битов записываются в виде строк из нулей и единиц в одинарных кавычках, например, '1000 0001'. Пробелы в строке битов предназначены для простоты чтения и не имеют никакого значения.

ch	Канал
gr	Блок из 3 * 32 суб-полосных отсчетов на уровне Аудио II, 18 * 32 суб-полосных отсчетов на уровне Аудио III.
main_data	Участок основных данных (main_data) двоичного потока содержит коэффициенты масштабирования, данные, кодированные по алгоритму Хаффмана, и дополнительную информацию.
main_data_beg	Указывает на то, что данный кадр расположен внутри двоичного потока в начале поля main_data. Это положение соответствует положению окончания поля main_data предыдущего кадра плюс 1 бит. Оно вычисляется из значения main_data_end предыдущего кадра.
part2_length	Это значение содержит данные о числе битов main_data, использованных для описания коэффициентов масштабирования.
rpchof	Полиномиальные коэффициенты остатка, начиная с наибольшего
sb	Субполоса
scfsi	Информация о селекторе коэффициента масштабирования
switch_point_l	Номер полосы коэффициента масштабирования (полосы коэффициента масштабирования длинного блока), в этой точки окна используется переключение
switch_point_s	Номер полосы коэффициента масштабирования (полосы коэффициента масштабирования короткого блока), в этой точки окна используется переключение
tcimsbf	Целое число с дополнением до двух, старший бит (знак) первый
uimsbf	Целое число без знака, старший бит первый
vlclbf	Код переменной длины, Строка битов, в которой левый бит является первым, где "левый" означает порядок, в котором записаны коды переменной длины
window	номер данного временного интервала в случае, когда block_type = = 2, $0 \leq$ window ≤ 2.

Порядок следования байтов в многобайтовых словах – старший бит первый .

2.2.7 Постоянные

π	3.14159265359
e	2.71828182845

2.3 Метод описания синтаксиса двоичного потока

Восстановленные декодером двоичные потоки описываются в § 2.4.1 и § 2.5.1. Каждый блок данных в двоичном потоке напечатан жирным шрифтом. Он описывается своим названием, длиной в битах, мнемокодом своего типа и порядком передачи.

Действие, выполняемое элементом декодированных данных в двоичном потоке, зависит от значения этого элемента данных и от ранее декодированных элементов данных. Декодирование элементов данных и определение состояния переменных, использованных при их декодировании, описываются в разделах, содержащих семантическое описание синтаксис. Нижеприведенные конструкции используются для описания условий, в которых элементы данных представлены и имеют нормальный тип.

Описание этого синтаксиса выполнено с использованием условных обозначений кодов "C", в соответствии с которыми, если переменная или выражение имеет ненулевое значение, то это означает, что условие выполняется (имеет значение "true"):

while (условие) {
data_element
. . .
}
	Если это условие выполняется, то далее в потоке данных появляется группа элементов данных **data_element**. Это повторяется до тех пор, пока это условие не перестанет выполняться.

do {
data_element
. . .
}
while (условие)
	Элемент данных всегда появляется, как минимум, один раз. Этот элемент данных повторяется до тех пор, пока не перестанет выполняться это условие.

if (условие) {
data_element
. . .
}
	Если это условие выполняется, то далее в потоке данных появляется первая группа элементов данных **data_element**.

else {
data_element
. . .
}
	Если это условие не выполняется, то далее в потоке данных появляется вторая группа элементов данных **data_element**.

for (i = 0; i < n; i++) {
data_element
. . .
}
	Группа элементов данных появляется n раз. Условные конструкции внутри группы элементов данных могут зависеть от значения переменной i управления циклом, которая для первого появления устанавливается равной нулю, и увеличивается на 1 при втором появлении и так далее.

Как отмечено, группа элементов данных может содержать вложенные условные конструкции. Для более компактного описания, скобки {} опускаются, если в них содержится только один элемент данных:

data_element []	data_element [] – это массив данных. Количество элементов данных указывается содержанием.
data_element [n]	data_element [n] – это n+1-й элемент массива данных.
data_element [m][n]	data_element [m][n] – это m+1-й, n+1-й элемент двумерного массива данных.
data_element [l][m][n]	data_element [l][m][n] – это l+1-й, m+1-й, n+1-й элемент трехмерного массива данных.
data_element [m..n]	Это диапазон битов между битом №m и битом №n ("включительно") в элемента денных data_element.

Хотя синтаксис выражается в процедурных терминах, не следует предполагать, что реализация схем, показанных на рисунке 2-1 или 2-2, позволит получить удовлетворительную процедуру декодирования. В частности, они определяют правильный входящий поток данных и входящий поток данных без ошибок. Реальные декодеры должны включать в себя средства для поиска стартовых кодов и байтов синхронизации (Транспортный поток), для того чтобы начать декодирование правильным образом и для того чтобы определить в процессе декодирования ошибки, потери или введенные лишние биты. Методы для определения таких ситуаций и действия, которые должны быть предприняты, не стандартизуются.

2.4 Требования к двоичному транспортному потоку

2.4.1 Структура кодирования и параметры Транспортного потока

Уровень кодирования Транспортного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, позволяет объединять одну или несколько программ в единый поток. Данные каждого элементарного потока мультиплексируются вместе с информацией, которая позволит выполнить синхронизированное воспроизведение элементарных потоков в рамках программы.

Транспортный поток состоит из одной или нескольких программ. Аудио и видео элементарные потоки состоят из модулей доступа.

Данные элементарных потоков передаются внутри пакетов PES. Пакет PES состоит из заголовка пакета PES, за которым следуют данные пакета. Пакеты PES вводятся в пакеты Транспортного потока. Первый байт каждого заголовка пакета PES находится в первом доступном свободном месте пакета Транспортного потока.

Заголовок пакета PES начинается с 32-битового стартового кода, который также идентифицирует проток или тип потока, которому принадлежат данные пакета. Заголовок пакета PES может содержать метки времени декодирования и воспроизведения (DTS и PTS). Заголовок пакета PES также содержит другие дополнительные поля. Поле данные пакета PES содержит переменное количество непрерывных байтов одного элементарного потока.

Пакеты Транспортного потока начинаются с 4-байтового префикса, который содержит 13-битовый идентификатор пакета (PID), определенный в таблице 2-2. Идентификатор PID определяет, посредством таблиц Специальной информации программы (PSI), содержание данных, находящихся в пакете Транспортного потока. Пакеты Транспортного потока с одинаковым значением PID содержат данные одного и только одного элементарного потока.

Таблицы PSI передаются в транспортном потоке. Имеется шесть таблиц PSI:

- Таблица ассоциаций для программы (Program Association Table);
- Таблица преобразования программы (Program Map Table);
- Таблица условного доступа (Conditional Access Table);
- Таблица сетевой информации (Network Information Table);

- Таблица описания транспортного потока (Transport Stream Description Table);
- Таблица управляющей информации IPMP (IPMP Control Information Table).

Эти таблицы содержат необходимую и достаточную информацию для демультиплексирования и воспроизведения программ. Таблица преобразования программы в таблице 2-33 наряду с прочей информацией указывает, какие PID и, следовательно, какие элементарные потоки используются для формирования каждой программы. Эта таблица также указывает PID пакетов транспортного потока, которые передают PCR для каждой программы. Таблица условного доступа должна присутствовать, если используется скремблирование. Таблица сетевой информации является необязательной, ее содержание не определяется в настоящей Рекомендации | Международном стандарте. Таблица управляющей информации IPMP должна присутствовать, если какой-либо из компонентов в потоке по Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК13818-11 использует IPMP, как описано в стандарте ИСО/МЭК13818-1.

Пакеты Транспортного потока могут быть нулевыми пакетами. Нулевые пакеты предназначены для заполнения Транспортных потоков. Они должны быть введены или удалены при помощи процессов повторного мультиплексирования и, следовательно, не следует предполагать доставку на декодер полезной нагрузки из нулевых пакетов.

В настоящей Рекомендации | Международном стандарте не определяются кодированные данные, которые могут использоваться как часть систем условного доступа. Однако в настоящей Спецификации для поставщиков услуг программ описываются механизмы транспортировки и идентификации этих данных для обработки в ходе декодирования, и для правильного указания данных, которые определяются в настоящей Спецификации. Такой тип поддержки обеспечивается как за счет структуры пакета Транспортного потока, так и в таблице условного доступа (см. таблицу 2-32 для PSI).

2.4.2 Декодер конечной системы Транспортного потока

Семантика Транспортного потока, определенная в § 2.4.3, и ограничения этой семантики, определенные в § 2.7, требуют точного определения событий прибытия и декодирования, а также моментов времени, в которые они происходят. Необходимые определения устанавливаются в настоящей Рекомендации | Международном стандарте с использованием гипотетического декодера, известного под названием Декодера конечной системы Транспортного потока (T-STD). Подробное описание T-STD содержится в информативном Приложении D.

Декодер конечной системы Транспортного потока (T-STD) – это концептуальная модель, используемая для точного определения этих терминов и для моделирования процесса декодирования во время создания или проверки Транспортных потоков. Декодер T-STD определяется только для этих целей. В T-STD существует три типа декодеров: видеодекодер, аудиодекодер и системный декодер. Пример показан на рисунке 2-1. Ни архитектура декодера T-STD, ни описанный процесс синхронизации не препятствует непрерывному, синхронному воспроизведению Транспортных потоков на выходе разнообразных декодеров, имеющих различные архитектуры или правила синхронизации.

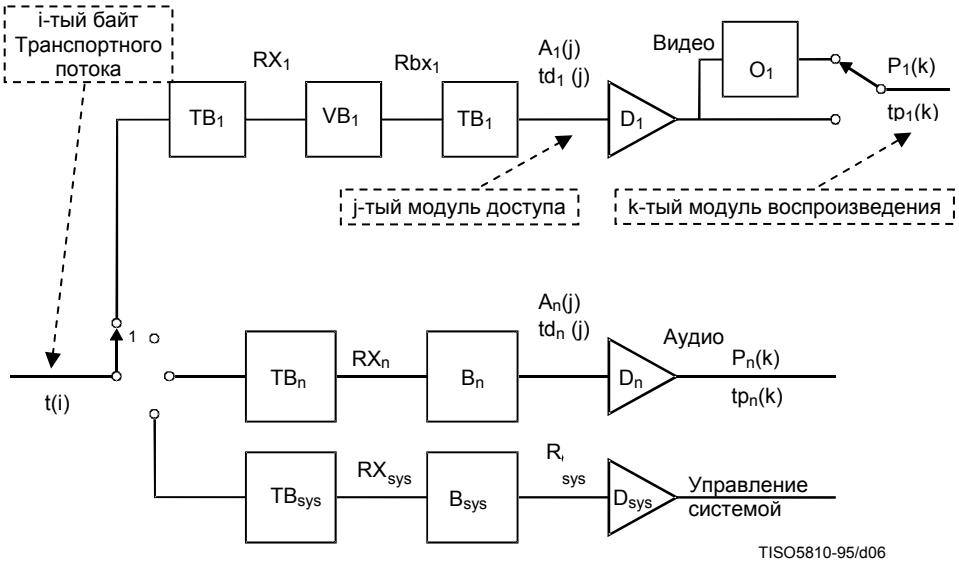


Рисунок 2-1 – Нотация декодера конечной системы Транспортного потока

Нижеприведенная нотация используется для описания декодера конечной системы и частично проиллюстрирована на рисунке 2-1, выше.

i, i', i'' индексы байтов в Транспортном потоке. Первый байт имеет индекс 0.

j индекс модулей доступа в элементарных потоках.

k, k', k'' индексы модулей представления в элементарных потоках.

n индекс элементарных потоков.

p индекс пакетов Транспортного потока в Транспортном потоке.

$t(i)$ указывает время в секундах, в которое i-тый байт Транспортного потока поступает на декодер конечной системы. Значение $t(0)$ – произвольная постоянная.

$PCR(i)$ время, закодированное в поле PCR, измеренное в единицах периодов системных часов, работающих с частотой 27 МГц, где i – индекс байта последнего байта поля program_clock_reference_base.

$A_n(j)$ j-тый модуль доступа в элементарном потоке "n". Модули $A_n(j)$ нумеруются в порядке декодирования.

$td_n(j)$ измеренное в секундах время декодирования декодером конечной системы для j-того модуля доступа в элементарном потоке "n".

$P_n(k)$ k-тый модуль представления в элементарном потоке "n". $P_n(k)$ нумеруются в порядке в воспроизведения.

$tp_n(k)$ измеренное в секундах время воспроизведения на декодере конечной системы k-того модуля представления в элементарном потоке "n".

t время, измеренное в секундах.

$F_n(t)$ измеренная в байтах степень наполнения входного буфера декодера конечной системы для элементарного потока "n" в момент времени t.

B_n основной буфер для элементарного потока "n". Он представлен только для аудио элементарных потоков.

BS_n размер буфера B_n, измеренный в байтах.

B_{sys} основной буфер декодера конечной системы для системной информации программы, которая находится в процессе декодирования.

BS_{sys} размер буфера B_{sys}, измеренный в байтах.

MB_n буфер мультиплексирования для элементарного потока "n". Он представлен только для видео элементарных потоков.

MBS_n размер буфера MB_n, измеренный в байтах.

EB_n буфер элементарного потока для элементарного потока "n" Он представлен только для видео элементарных потоков.

EBS_n размер буфера элементарного потока EB_n, измеренный в байтах.

TB_{sys} транспортный буфер для системной информации программы, которая находится в процессе декодирования.

TBS_{sys} размер буфера TB_{sys}, измеренный в байтах.

TB_n транспортный буфер для элементарного потока "n".

TBS_n размер буфера TB_n, измеренный в байтах.

D_{sys} декодер для системной информации в Программном потоке "n".

D_n декодер для элементарного потока "n".

O_n буфер изменения порядка для видео элементарного потока "n".

R_{sys} скорость, с которой данные удаляются из B_{sys}.

Rx_n скорость, с которой данные удаляются из TB_n.

Rbx_n скорость, с которой данные полезной нагрузки пакета PES удаляются из MB_n, когда используется метод утечки. Определяется только для видео элементарных потоков.

$Rbx_n(j)$ скорость, с которой данные полезной нагрузки пакета PES удаляются из MB_n, когда используется метод vbv_delay. Определяется только для видео элементарных потоков.

Rx_{sys} скорость, с которой данные удаляются из TB_{sys}.

R_{es} скорость видео элементарного потока, кодированные в заголовке последовательности.

2.4.2.1 Частота системных часов

Информация синхронизации, используемая в T-STD в качестве опорной, передается в нескольких полях данных, определенных в настоящей Спецификации. См. § 2.4.3.4 и 2.4.3.6. Эта информация кодируется в полях PCR в виде отсчетов с частотой системных часов программы. Поля PCR также передаются в полях адаптации пакетов Транспортного потока, имеющих значение PID, равное PCR_PID, определенное в секции TS_program_map_section декодируемой программы.

Реальные декодеры могут восстановить частоту системных часов программы из этих значений и их соответствующего времени прибытия. Далее приведены минимальные ограничения, которые применяются к частоте системных часов, представленные значениями полей PCR, когда они принимаются на декодере.

Значение частоты системных часов измеряется в Герцах и должно удовлетворять следующим ограничениям:

– 27 000 000 – 810 <= system_clock_frequency <= 27 000 000 + 810;

– Скорость изменения частоты системных часов system_clock_frequency во времени $\leq 75 \times 10^{-3}$ Гц/с

ПРИМЕЧАНИЕ. – Источники кодированных данных должны иметь небольшие допуски, для того чтобы облегчить корректную работу устройств записи и воспроизведения потребителя.

Частота системных часов (system_clock_frequency) программы может быть более точной, чем требуется. О такой повышенной точности может быть сообщено на декодер при помощи дескриптора Системных часов, которые описан в § 2.6.20.

Скорости передачи, определенные в настоящей Спецификации, измеряются относительно единиц частоты system_clock_frequency. Например, скорость передачи = 27 000 000 битов в секунду в T-STD будет указывать, что каждые восемь (8) циклов системных часов передается один байт данных.

Нотация "system_clock_frequency" используется в нескольких местах настоящей Спецификации для указания частоты системных часов, соответствующей этим требованиям. Для удобства обозначения, уравнения, в которых встречаются значения PCR, PTS или DTS, приводят к получению значений времени, выраженных с точностью до некоторого целого числа секунд, кратного (300×2^{33}/system_clock_frequency). Это происходит в результате кодирования информации синхронизации PCR в виде 33 битов с частотой дискретизации = 1/300 от частоты системных часов плюс 9 битов на остаток, и кодирование в виде 33 битов с частотой системных часов, поделенной на 300 для PTS и DTS.

2.4.2.2 Входной сигнал для декодера конечной системы Транспортного потока

Входной сигнал для декодера конечной системы Транспортного потока (T-STD) представляет собой Транспортный поток. Транспортный поток может содержать несколько программ с независимыми временными базами. Однако в каждый момент времени T-STD декодирует только одну программу. В модели T-STD все указания синхронизации ссылаются на временную базу этой программы.

Данные Транспортного потока поступают на T-STD с фрагментарно-постоянной скоростью. Момент времени t(i), когда i-тый байт поступает на T-STD, определяется путем декодирования полей эталонного времени программы (PCR) входного потока, закодированных в поле адаптации Транспортного потока программы, которая должна быть декодирована, и путем подсчета байтов в полном Транспортном потоке между последовательными полями PCR этой программы. Поле PCR (см. уравнение 2-1) кодируется в виде двух частей: одной в единицах периода частоты, равной 1/300 от частоты системных часов, которая называется program_clock_reference_base (см. уравнение 2-2), и одной в единицах частоты системных часов, которая называется program_clock_reference_extension (см. уравнение 2-3). Значения, закодированные в этих частях, вычисляются в виде PCR_base(i) (см. уравнение 2-2) и PCR_ext(i) (см. уравнение 2-3) соответственно. Значение, закодированное в поле PCR, указывает время t(i), где i – индекс байта, содержащего последний бит поля program_clock_reference_base.

В частности:

$$PCR(i) = PCR_base(i) \times 300 + PCR_ext(i)\,, \tag{2-1}$$

где:

$$PCR_base(i) \;=\; ((system_clock_frequency \times t(i))\, DIV\; 300)\; \%\; 2^{33} \tag{2-2}$$

$$PCR_ext(i) \;=\; ((system_clock_frequency \;\times t(i))\, DIV\; 1)\; \%\; 300 \tag{2-3}$$

Для всех других байтов время прихода входного сигнала t(i), показанное в уравнении 2-4 ниже, вычисляется из PCR(i″) и скорости транспортировки, с которой прибывают данные, где скорость транспортировки определяется как число байтов в Транспортном потоке между байтами, содержащими последние биты двух последовательных полей program_clock_reference_base одной и той же программы, поделенное на разницу между значениями времени, закодированных в этих полях PCR.

$$t(i) = \frac{PCR(i'')}{system_clock_frequency} + \frac{i - i''}{transport_rate(i)}, \qquad (2\text{-}4)$$

где:

i: индекс любого байта в Транспортном потоке для i″ < i < i′.

i″: индекс байта, содержащего последний бит последнего поля program_clock_reference_base, применительно к декодируемой программе.

PCR(i″): время. в единицах системных часов, закодированное в полях эталонной временной базы программы и полях расширения эталонной временной базы.

Скорость транспортировки определяется выражением:

$$transport_rate(i) = \frac{((i - i'') \times system_clock_frequency)}{PCR(i') - PCR(i'')}, \qquad (2\text{-}5)$$

где:

i′: индекс байта, содержащего последний бит следующего поля program_clock_reference_base, применительно к декодируемой программе.

ПРИМЕЧАНИЕ. – i″ < i ≤ i′.

В случае нарушения непрерывности временной базы, что указано индикатором iscontinuity_indicator в поле адаптации транспортного пакета, между последним PCR старой временной базы и первым PCR новой временной базы выражение, описанное уравнением 2-4 и уравнением 2-5 для времени прибытия байтов на вход T-STD, неприменимо. В таком случае, время прибытия этих байтов определяется в соответствии с уравнением 2-4 с той модификацией, чтобы используемая в нем скорость транспортировки, была бы применима между последним PCR старой временной базы и следующим PCR старой временной базы.

Для значений PCR указывается допуск. Допуск PCR определяется как максимальная неточность, допускаемая в принимаемых значениях PCR. Эта неточность может быть обусловлена неточностью значений PCR или изменением PCR во время повторного мультиплексирования. Она не учитывает ошибки времени прибытия пакетов из-за нестабильности сети или из-за других причин. Допуск для PCR = ± 500 нс.

В модели T-STD неточность может быть выражена как неточность скорости транспортировки, вычисленной с использованием уравнения 2-5.

Транспортные потоки с несколькими программами и переменной скоростью

Транспортные потоки могут содержать несколько программ, имеющих независимые временные базы. Для каждой из этих независимых программ требуются раздельные наборы значений PCR, указанных соответствующими значениями PCR_PID, и, следовательно, значения PCR не могут располагаться совместно. Для программы, поступающей на вход T-STD, скорость передачи Транспортного потока является фрагментарно-постоянной. Следовательно, если скорость передачи Транспортного потока переменная, она может меняться только в полях PCR рассматриваемой программы. Поскольку значения PCR и, следовательно, точки в Транспортном потоке, где меняется скорость, расположены не вместе, скорость, с которой Транспортный поток поступает на T-STD, будет вынуждена изменяться в зависимости от того, какая программа поступает на T-STD. Следовательно, невозможно создать общее расписание доставки полного Транспортного потока на T-STD, когда этот Транспортный поток содержит несколько программ с независимыми временными базами, и скорость Транспортного потока переменная. Однако достаточно легко создать Транспортные потоки с постоянной скоростью из нескольких программ с переменной скоростью.

2.4.2.3 Буферизация

Полные пакеты транспортного потока, содержащие системную информацию для программы, выбранной для дешифрации, поступают в транспортный буфер системы TB_{sys} со скоростью транспортного потока. Они включают пакеты транспортного потока со значениями PID 0, 1, 2 или 3 и все пакеты транспортного потока, идентифицированные в Таблице ассоциаций для программы (см. таблицу 2-30) как пакеты, имеющие значение идентификатора program_map_PID для выбранной программы. Данные Таблицы сетевой информации (NIT), определенные NIT PID, не передаются в TB_{sys}.

ПРИМЕЧАНИЕ 1. – Размер таблицы управляющей информации IPMP может быть большим, и частота повторения этой таблицы должна быть скорректирована с учетом требований к буферу.

Все байты, которые поступают в буфер TB_n, удаляются со скоростью Rx_n, определенной ниже. Байты, которые являются частью пакета PES или его содержимое, доставляются в основной буфер B_n для аудио элементарных потоков и системных данных, и в буфер мультиплексирования MB_n для видео элементарных потоков. Другие туда не доставляются, и они могут использоваться для управления системой. Дубликаты пакетов Транспортного потока не доставляются в B_n, MB_n и B_{sys}.

Буфер TB_n очищается следующим образом:

– Когда в TB_n нет данных, Rx_n равно нулю.

– В ином случае для видео:

$$Rx_n = 1,\ 2 \times R_{\max}[profile, level],$$

где:

R_{max}[profile, level] определяется в соответствии с профилем и уровнем, которые можно найти в таблице 8-13 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2. В этой Таблице определяется верхняя граница скорости каждого элементарного видеопотока в пределах конкретного профиля и уровня.

Rx_n равно 1, 2 × R_{max} для видеопотоков с ограниченными параметрами, соответствующих ИСО/МЭК 11172-2, где R_{max} обозначает максимальную скорость передачи для двоичного потока с ограниченными параметрами в соответствии с ИСО/МЭК 11172-2.

Для аудио потоков ADTS, соответствующих ИСО/МЭК 13818-7:

Количество каналов	**Rx_n [бит/с]**
1-2	2 000 000
3-8	5 529 600
9-12	8 294 400
13-48	33 177 600

Каналы: число выходных аудиоканалов с полной шириной полосы плюс некоторое количество независимо коммутируемых элементов соединения каналов в пределах одного и того же элементарного аудиопотока. Например, в типичном случае, когда независимо коммутируемых элементов соединения каналов нет, монофоническая передача – занимает один канал, стереофоническая – два канала, а канал объемного звучания 5.1 занимает 5 каналов (канал низкочастотных эффектов (LFE) не учитывается).

Для других аудиопрограмм:

$$Rx_n = 2 \times 10^6 \text{ битов в секунду}$$

Для системных данных:

$$Rx_n = 1 \times 10^6 \text{ битов в секунду}$$

Rx_n измеряется в отношении частоты системных часов.

Полные пакеты Транспортного потока, содержащие системную информацию программы, выбранной для декодирования, поступают в транспортный буфер системы TB_{sys} со скоростью передачи Транспортного потока. В эти пакеты входят пакеты Транспортного потока, у которых значения PID равны 0, 1, 2 и 3 (если представлены), и все пакеты Транспортного потока, идентифицированные при помощи Таблицы ассоциаций для программы (см. таблицу 2-30), как имеющие значение program_map_PID для выбранной программы. Данные Таблицы сетевой информации (NIT), определенные идентификатором NIT PID, в TB_{sys} не передаются.

Байты удаляются из TB_{sys} со скоростью Rx_{sys} и доставляются в B_{sys}. Каждый байт передается немедленно.

Дубликаты пакетов Транспортного потока в B_{sys} не доставляются.

Транспортные пакеты, которые не поступают ни в TB_n, ни в TB_{sys}, отбрасываются.

Размер транспортного буфера фиксирован и равен 512 байтов.

Размеры буферов элементарного потока EBS_1 – EBS_n определяются для видеопотока как равные значению поля vbv_buffer_size, которое передается в заголовке последовательности в заголовке последовательности. Подробности содержатся в главе "Обзор ограниченных параметров" стандарта ИСО/МЭК 11172-2 и в таблице 8-14 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2.

Размеры буферов мультиплексирования MBS_1–MBS_n определяются для видеопотока следующим образом:

Для Низкого и Основного уровня:

$$MBS_n = BS_{mux} + BS_{oh} + VBV_{\max}[profile, level] - vbv_buffer_size ,$$

где BS_{oh} – буферизация заголовка пакета PES определяется как:

$$BS_{oh} = (1/750)\,\text{seconds} \times R_{\max}[profile, level]$$

и BS_{mux} – дополнительная буферизация мультиплексирования определяется как:

$$BS_{mux} = 0{,}004\ \text{seconds} \times R_{\max}[profile, level]$$

и где VBV_{max}[profile, level] определяется в таблице 8-14 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, R_{max}[profile, level] определяется в таблице 8-13 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, а размер буфера vbv передается в заголовке последовательности, описанном в § 6.2.2 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2.

Для Высокого 1440 и Высокого уровня:

$$MBS_n = BS_{mux} + BS_{oh} ,$$

где BS_{oh} определяется следующим образом:

$$BS_{oh} = (1/750)\ \text{seconds} \times R_{\max}[profile, level] ,$$

BS_{mux} определяется следующим образом:

$$BS_{mux} = 0{,}004\,\text{seconds} \times R_{\max}[profile, level]$$

и где R_{max}[profile, level] определяется в таблице 8-13 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2.

Для потоков с ограниченными параметрами, соответствующими стандарту ИСО/МЭК 11172-2:

$$MBS_n = BS_{mux} + BS_{oh} + vbv_\max - vbv_buffer_size$$

где BS_{oh} определяется следующим образом:

$$BS_{oh} = (1/750)\,\text{seconds} \times R_{\max} ,$$

BS_{mux} определяется следующим образом:

$$BS_{mux} = 0{,}004\,\text{seconds} \times R_{\max}$$

и где R_{max} и vbv_max указывают максимальную скорость передачи и максимальное значение размера vbv_buffer_size для потока с ограниченными параметрами, соответствующего стандарту ИСО/МЭК 11172-2, соответственно.

Часть BS_{mux} = 4 мс × R_{max}[profile, level] размера буфера MBS_n распределена для буферизации, которая позволяет выполнять мультиплексирование. Оставшаяся часть доступна для BS_{oh} и может быть также доступна для первоначального мультиплексирования.

ПРИМЕЧАНИЕ 2. – Занятость буфера заголовком пакета PES ограничена в потоках PES непосредственно декодером PES-STD, который определен в § 2.5.2.4. Допустимо, но не необходимо использовать потоки PES для создания Транспортных потоков.

Буфер BS_n

Размеры основных буферов BS_1–BS_n определяются следующим образом.

Аудио

Для аудиопотока ADTS, соответствующего ИСО/МЭК 13818-7:

Количество каналов	BS_n [байтов]
1-2	3 584
3-8	8 976
9-12	12 804
13-48	51 216

Каналы: число выходных аудиоканалов с полной шириной полосы плюс некоторое количество независимо коммутируемых элементов соединения каналов в пределах одного и того же элементарного аудиопотока. Например, в типичном случае, когда нет независимо коммутируемых элементов соединения каналов, монофоническая передача – занимает один канал, стереофоническая – два канала, а канал объемного звучания 5.1 занимает 5 каналов (канал низкочастотных эффектов (LFE) не учитывается.

Для других аудиопрограмм:

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \ \textit{байтов}$$

Размеры буфера для декодирования модуля доступа BS_{dec}, и буфера для заголовка пакета PES BS_{oh} ограничены значением:

$$BS_{dec} + BS_{oh} \leq 2848 \ \textit{байтов}$$

Часть (736 байтов) буфера размером 3584 байтов распределена для буферизации, которая позволяет выполнять мультиплексирование. Оставшаяся часть – 2848 байтов делится между буферизацией модуля доступа (BS_{dec}) и буферизацией для дополнительного мультиплексирования (BS_{oh}).

Системы

Основной буфер B_{sys} для системных данных имеет размер BS_{sys} = 1536 байтов.

Видео

Данные элементарных видеопотоков передаются из буфера MB_n в буфер EB_n при помощи двух методов: метода утечки и метода задержки VBV.

Метод утечки

Метод утечки передает данные из буфера MB_n в буфер EB_n, используя скорость утечки R_{bx}. Метод утечки используется всегда, когда выполняются следующие условия:

- в Транспортном потоке не представлен дескриптор STD (см. § 2.6.32) для элементарного потока;
- дескриптор STD представлен, и флаг leak_valid имеет значение '1';
- дескриптор STD представлен, флаг leak_valid имеет значение '0', и поля vbv_delay, закодированные в видеопотоке имеют значение 0xFFFF; или
- состояние "хитрого" режима имеет значение "true" (см. § 2.4.3.7).

Для Низкого и Основного уровня:

$$Rbx_n = R_{\max}[profile, level]$$

Для Высокого 1440 и Высокого уровня:

$$Rbx_n = Min\{1{,}05 \times R_{es}, R_{\max}[profile, level]\}$$

Для потоков с ограниченными параметрами, соответствующими стандарту ИСО/МЭК 11172-2:

$$Rbx_n = 1,2 \times R_{\max},$$

где R_{max} – максимальная скорость передачи для потока с ограниченными параметрами, соответствующего стандарту ИСО/МЭК 11172-2.

Если в буфере MB_n есть данные полезной нагрузки пакета PES, и буфер EB_n не полон, то полезная нагрузка пакета PES передается из MB_n в EB_n со скоростью, равной Rbx_n. Если буфер EB_n полон, то данные из MB_n не удаляются. Когда байт данных передается из MB_n в EB_n, все байты заголовка пакета PES, которые находятся в MB_n и непосредственно предшествуют этому байту, немедленно удаляются и отбрасываются. Когда в MB_n нет данных полезной нагрузки пакета PES, данные из MB_n не удаляются. Все данные, которые поступают в буфер MB_n, покидают его. Все байты полезной нагрузки пакета PES поступают в EB_n сразу после выхода из MB_n.

Метод Vbv_delay

Метод vbv_delay точно определяет момент времени, в который каждый байт кодированной видеоинформации передается из MB_n в EB_n, используя значения vbv_delay, закодированные в элементарном видеопотоке. Метод vbv_delay используется всегда, когда в Транспортном потоке представлен дескриптор STD (см. § 2.6.32) для этого элементарного потока, флаг leak_valid в дескрипторе имеет значение '0', и поля vbv_delay, закодированные в этом видеопотоке, не равны 0xFFFF. Если какие-либо значения vbv_delay в видеопоследовательности не равны 0xFFFF, то ни одно поле vbv_delay в этой последовательности не должно быть равно 0xFFFF (см. ИСО/МЭК 11172-2 и Рекомендацию МСЭ-Т H.262 | ИСО/МЭК 13818-2).

Когда используется метод vbv_delay, последний байт стартового кода j-того видеокадра передается из MB_n в EB_n в момент времени $td_n(j)$ – vbv_delay(j), где $td_n(j)$ – это время декодирования j-того видеокадра, как определено выше, и vbv_delay(j) – это время задержки в секундах, указанное в поле vbv_delay j-того видеокадра. Передача байтов между завершающими байтами стартовых кодов последовательных видеокадров (включая последний байт второго стартового кода), в буфер EB_n выполняется с фрагментарно-постоянной скоростью $R_{bx}(j)$, которая определяется для каждого j-того видеокадра. В частности, скорость передачи в этот буфер $R_{bx}(j)$ определяется выражением:

$$R_{bx}(j) = NB(j)/(vbv_delay(j) - vbv_delay(j+1) + td_n(j+1) - td_n(j)) \qquad (2\text{-}6)$$

где NB(j) – количество байтов между завершающими байтами стартовых кодов (включая последний байт второго стартового кода) j-того и j+1-го видеокадров, за исключением байтов заголовка пакета PES.

ПРИМЕЧАНИЕ 3. – vbv_delay(j + 1) и $td_n(j + 1)$ могут иметь значения, которые отличаются от обычно ожидаемых значений для периодического воспроизведения видеокадров, если флаг low_delay в расширении видеопоследовательности установлен в значение '1'. В ходе изучения двоичного потока может оказаться невозможным определить точные значения.

Значение $R_{bx}(j)$, полученное из уравнения 2-6, должно быть меньше или равно R_{max}[profile, level] для элементарных потоков типа 0x02 (см. таблицу 2-34), где R_{max}[profile, level] определяется в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, и должно быть меньше или равно максимальной скорости передачи, допустимой для элементарных видеопотоков с ограниченными параметрами типа 0x01, см. ИСО/МЭК 11172-2.

Когда байт данных передается из MB_n в EB_n, все байты заголовка пакета PES, которые находятся в MB_n и непосредственно предшествуют этому байту, немедленно удаляются и отбрасываются. Все данные, которые поступают в буфер MB_n, покидают его. Все байты полезной нагрузки пакета PES поступают в EB_n сразу после выхода из MB_n.

Удаление модулей доступа

Для каждого буфера элементарного потока EB_n и основного буфера B_n все данные модуля доступа, которые находятся в буфере дольше остальных $A_n(j)$, и все байты заполнения, которые ему непосредственно предшествуют, и которые находятся в буфере на момент времени $td_n(j)$, в момент времени $td_n(j)$ немедленно удаляются. Время декодирования $td_n(j)$ указывается в полях DTS или PTS (см. § 2.4.3.6). Значения времени декодирования $td_n(j + 1)$, $td_n(j + 2)$, ... модулей доступа без кодированных полей DTS или PTS, которые расположены сразу же после модуля доступа j, могут быть получены из информации, передаваемой в элементарном потоке. См. Приложение C Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, ИСО/МЭК 13818-3, или ИСО/МЭК 11172. Кроме того, см. § 2.7.5. Для аудиопотока все заголовки пакетов PES, которые хранятся непосредственно перед модулем доступа, или которые введены внутрь данных модуля доступа, удаляются одновременно с удалением модуля доступа. Когда модуль доступа удаляется, он немедленно декодируется, превращаясь в модуль представления.

Системные данные

В случае системных данных, данные удаляются из основного буфера B_{sys} со скоростью R_{sys} всегда, когда в буфере B_{sys} имеется хотя бы один байт.

$$R_{sys} = \max\,(80\,000 \text{ бит/с},\ transport_rate(i) \times 8\ \textit{битов} / \textit{байт} / 500) \qquad (2\text{-}7)$$

ПРИМЕЧАНИЕ 4. – Стремление увеличить R_{sys} в случае высокой скорости транспортировки обусловлена стремлением дать возможность увеличить скорость передачи данных для специальной информации программы.

Малая задержка

Когда флаг малой задержки (low_delay) в расширении видеопоследовательности установлен в значение '1' (см. § 6.2.2.3 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2) буфер EB_n может быть недозаполнен. В таком случае, когда буфер EB_n в T-STD элементарного потока проверяется в момент времени, указанный значением $td_n(j)$, в буфере EB_n может не оказаться полных данных модуля доступа. Когда такое случается, буфер должен повторно проверяться с интервалами, равными двум периодам поля, до тех пор, пока в буфере не будут представлены полные данные модуля доступа. В этот момент времени модуль доступа должен быть сразу целиком удален из буфера EB_n. Не должно возникать переполнения буфера EB_n.

Когда флаг low_delay_mode установлен в значение '1', допускается, чтобы буфер EB_n постоянно недозаполнялся без каких-либо ограничений. Декодер T-STD должен удалять данные модуля доступа из буфера EB_n сразу, как только это допустимо в соответствии с вышеприведенным параграфом и любыми значениями меток времени DTS или PTS, закодированными в двоичном потоке. Отметим, что декодер может не иметь возможности восстановить корректные значения времени декодирования и воспроизведения, указанные в DTS и PTS до тех пор, пока не завершится ситуация с недозаполнением буфера EB_n, и в двоичном потоке не будут найдено значение PTS или DTS.

"Хитрый" режим

Когда флаг "хитрого" режима DSM_trick_mode (2.4.3.6) установлен в '1', в заголовке пакета PES, содержащего начало видео модуля доступа типа B, а поле trick_mode_control установлено в '001' (замедленное воспроизведение) или '010' (стоп-кадр), или '100' (замедленное обратное воспроизведение), модуль доступа кадра B из буфера видеоданных EB_n не удаляется до последнего раза из возможного множества раз декодирования и воспроизведения любого поля данного кадра изображения. Повторение воспроизведения полей и кадров определяется в § 2.4.3.8 в режиме медленного воспроизведения, медленного обратного воспроизведения и field_id_cntrl. Модуль доступа удаляется из EB_n сразу в указанное время, которое зависит от значения поля rep_cntrl.

Когда флаг "хитрого" режима DSM_trick_mode установлен в '1' в заголовке пакета PES, содержащего первый байт стартового кода кадра, статус trick_mode принимает значение "true", когда из буфера EB_n удаляется стартовый код этого кадра в пакете PES. Статус "хитрого" режима остается в значении "true" до тех пор, пока заголовок пакета PES не будет принят декодером T-STD, в котором флаг DSM_trick_mode установлен в '0', первый байт стартового кода этого кадра после заголовка пакета PES не будет удален из буфера EB_n. Когда статус "хитрого" режима имеет значение "true", буфер EB_n может быть недозаполненным. Когда статус "хитрого" режима имеет значение "true", все остальные ограничения, установленные для обычных потоков, остаются неизменными.

2.4.2.4 Декодирование

Элементарные потоки, сохраняемые в буферах $B_1 – B_n$ и $EB_1 – EB_n$, немедленно декодируются декодерами $D_1 – D_n$, и могут быть задержаны в буферах переупорядочивания $O_1 – O_n$ прежде, чем поступить на выход декодера T-STD. Буферы переупорядочивания используются только для элементарного видеопотоков том случае, когда некоторые блоки доступа передаются в порядке, отличающемся от порядка воспроизведения. Порядок следования этих модулей доступа должен быть изменен до воспроизведения. В частности, если $P_n(k)$ представляет собой кадр I или кадр P, передаваемый перед одним или несколькими кадрами B, то он, до воспроизведения, должен быть задержан в буфере переупорядочивания O_n декодера T-STD. Любой кадр, который был ранее сохранен в буфере O_n, воспроизводится до того, как текущий кадр может быть сохранен. Кадр $P_n(k)$ должен быть задержан до того момента, как будет выполнено декодирование следующего кадра I или кадра P. Пока он хранится в буфере переупорядочивания, декодируются и воспроизводятся следующие за ним кадры B.

Момент времени, в который воспроизводится модуль представления $P_n(k)$, соответствует значению $tp_n(k)$. Для модулей представления, которым задержка на переупорядочивание не требуется, $tp_n(k) = td_n(j)$, поскольку модули доступа декодируются незамедлительно. Это, например, справедливо для кадров B. Для модулей представления, которые задерживаются, $tp_n(k)$ и $td_n(j)$ отличаются друг от друга на время, на которое $P_n(k)$ задерживается в буфере переупорядочивания, и которое кратно номинальному периоду кадров. Следует

проявлять осторожность и для того чтобы выполнялись требования ко всему потоку в целом, использовать адекватную задержку на переупорядочивание, начиная с начала элементарных видеопотоков. Например, для потока, который изначально содержит только кадры I и P, но позднее включает в себя кадры B, необходимо с самого начала потока предусмотреть задержку на переупорядочивание.

В Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 переупорядочивание видеокадров разъясняется более подробно.

2.4.2.5 Воспроизведение

Задачей системы декодирования является восстановление модулей представления из компрессированных данных и воспроизведение их в виде синхронизированной последовательности в правильные моменты времени воспроизведения. Хотя существующие устройства воспроизведения аудиовизуальной информации, как правило, обладают конечной и определенной задержкой, а также могут вносить дополнительные задержки, обусловленные выполнением функций пост-обработки или выходных функций, декодер конечной системы моделирует эти задержки, как равные нулю.

В декодере T-STD, показанном на рисунке 2-1, воспроизведение модуля представления визуальной информации (кадра) происходит незамедлительно в момент времени его воспроизведения $tp_n(k)$.

В декодере T-STD воспроизведение модуля представления звуковой информации начинается в момент времени его воспроизведения $tp_n(k)$, когда декодер незамедлительно воспроизводит первый отсчет. Последующие отсчеты модуля представления воспроизводятся последовательно со скоростью, соответствующей частоте дискретизации звукового сигнала.

2.4.2.6 Управление буфером

Транспортные потоки должны быть созданы таким образом, чтобы выполнялись условия, определенные в настоящем подразделе. В настоящем подразделе используется нотация, определенная для декодера конечной системы.

Буферы TB_n и TB_{sys} не должны переполняться. Буферы TB_n и TB_{sys} должны освобождаться, как минимум, раз в секунду. Буфер B_n не должен ни переполняться, ни оставаться недозаполненным. Буфер B_{sys} не должен переполняться.

Буфер EB_n не должен оставаться недозаполненным, за исключением случая, когда флаг малой задержки в расширении видеопоследовательности установлен в '1' (см. § 6.2.2.3 в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2) или поле статуса trick_mode имеет значение "true".

Когда для определения передачи используется метод утечки, буфер MB_n не должен переполняться, и должен освобождаться, как минимум, раз в секунду. Буфер EB_n не должен переполняться.

Когда для определения передачи используется метод vbv_delay, буфер MB_n не должен ни переполняться, ни оставаться недозаполненным, и буфер EB_n не должен переполняться.

Задержка любых данных в буферах декодера конечной системы должна быть меньше или равна одной секунде, за исключением случаев передачи видеоданных с неподвижными изображениями и потоков, соответствующих ИСО/МЭК 14496. В частности: $td_n(j) - t(i) \leq 1$ секунде для всех значений j и для всех i-тых байтов в модуле доступа $A_n(j)$.

Для случаев передачи видеоданных с неподвижными изображениями, задержка ограничивается значением $td_n(j) - t(i) \leq 60$ секунд для всех значений j и для всех байтов i в модуле доступа $A_n(j)$.

Для потоков, соответствующих ИСО/МЭК 14496, задержка ограничивается значением $td_n(j) - t(i) \leq 10$ секунд для всех значений j и для всех байтов i в модуле доступа $A_n(j)$.

Определение переполнения и недозаполнения

Пусть $F_n(t)$ – степень заполнения буфера B_n декодера T-STD в данный момент времени.

$F_n(t) = 0$ непосредственно перед моментом времени $t = t(0)$

Переполнение не возникает, если для всех значений t и n:

$$F_n(t) \leq BS_n$$

Недозаполнения не возникает, если для всех значений t и n:

$$0 \leq F_n(t)$$

2.4.2.7 Расширения декодера T-STD для передачи данных, соответствующих ИСО/МЭК 14496

Для декодирования данных, соответствующих ИСО/МЭК 14496, передаваемых в Транспортном потоке, модель T-STD расширяется. Параметры декодера T-STD для декодирования отдельных элементарных потоков, соответствующих ИСО/МЭК 14496, определяются в § 2.11.2, тогда как в § 2.11.3 определяются расширения T-STD и параметры декодирования изображений, соответствующих ИСО/МЭК 14496, и связанных с ними потоков.

2.4.2.8 Расширение T-STD для передачи видеоданных, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10

Для того чтобы определить соответствующее Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 декодирование T-STD для видеопотоков, передаваемых в транспортном потоке, модель T-STD должна быть расширена. Расширение T-STD и параметры T-STD для декодирования видеопотоков соответствующих Рекомендации МСЭ-Т. H.264 | ИСО/МЭК 14496-10, определены в § 2.14.3.1.

2.4.3 Спецификация синтаксиса и семантики Транспортного потока

Нижеприведенный синтаксис описывает поток байтов. Пакеты Транспортного потока должны иметь длину 188 байтов.

2.4.3.1 Транспортный поток

См. таблицу 2-1.

Таблица 2-1 – Транспортный поток

Синтаксис	Количество битов	Мнемоника

MPEG_transport_stream() {
do {
transport_packet()
} while (nextbits() = = sync_byte)
}		

2.4.3.2 Уровень пакетов Транспортного потока

См. таблицу 2-2.

Таблица 2-2 – Пакет Транспортного потока, соответствующий настоящей Рекомендации | Международному стандарту

Синтаксис	Количество битов	Мнемоника
transport_packet(){		
sync_byte	**8**	**bslbf**
transport_error_indicator	**1**	**bslbf**
payload_unit_start_indicator	**1**	**bslbf**
transport_priority	**1**	**bslbf**
PID	**13**	**uimsbf**
transport_scrambling_control	**2**	**bslbf**
adaptation_field_control	**2**	**bslbf**
continuity_counter	**4**	**uimsbf**
if(adaptation_field_control = = '10' || adaptation_field_control = = '11'){		
adaptation_field()		
}		
if(adaptation_field_control = = '01' || adaptation_field_control = = '11') {		
for (i = 0; i < N; i++){		
data_byte	**8**	**bslbf**
}		
}		
}		

2.4.3.3 Определение семантики полей на уровне пакетов Транспортного потока

sync_byte – Поле sync_byte – это 8-битовое поле фиксированной длины, значение которого равно '0100 0111' (0x47). Следует избегать моделирования поля sync_byte при выборе значений для других регулярно появляющихся полей, например, таких как PID.

transport_error_indicator – Индикатор transport_error_indicator – это 1-битовый флаг. Когда он установлен в '1', он указывает, что в соответствующем пакете Транспортного потока существует, как минимум, 1 бит с неисправимой ошибкой. Этот бит может быть установлен в значение '1' средствами, которые являются внешними по отношению к уровню транспортировки. Когда этот бит установлен в '1', он не должен переустанавливаться в '0', если только значение(я) ошибочного(ых) бита(ов) не было(и) исправлено(ы).

payload_unit_start_indicator – Индикатор payload_unit_start_indicator – это 1-битовый флаг, который имеет нормативное значение для пакетов Транспортного потока, в которых передаются пакеты PES (см. § 2.4.3.6) или данные PSI (см. § 2.4.4).

Когда полезная нагрузка пакета Транспортного потока содержит данные пакета PES, индикатор payload_unit_start_indicator имеет следующее значение: '1' указывает, что полезная нагрузка этого пакета Транспортного потока будет начинаться в первом байте пакета PES, и '0' указывает, что в этом пакете Транспортного потока не должен начинаться ни один пакет PES. Если индикатор payload_unit_start_indicator установлен в '1', то этом пакете Транспортного потока начинается один и только один пакет PES. Это справедливо также для конфиденциальных потоков типа stream_type 6 (см. таблицу 2-34).

Когда полезная нагрузка пакета Транспортного потока содержит данные PSI, индикатор payload_unit_start_indicator имеет следующее значение: если пакет Транспортного потока содержит первый байт секции PSI, то значение индикатора payload_unit_start_indicator должно быть '1', указывая, что первый байт полезной нагрузки этого пакета Транспортного потока содержит поле pointer_field. Если пакет Транспортного потока не содержит первого байта секции PSI, то значение индикатора payload_unit_start_indicator должно быть '0', указывая, что в полезной нагрузке нет поля pointer_field. См. § 2.4.4.1 и § 2.4.4.2. Это справедливо также для конфиденциальных потоков типа stream_type 5 (см. таблицу 2-34).

Для нулевых пакетов индикатор payload_unit_start_indicator должен быть установлен в '0'.

Значение этого бита для пакетов Транспортного потока, содержащих только конфиденциальную информацию, в настоящей Спецификации не определяется

transport_priority – Индикатор transport_priority – это 1-битовый индикатор. Когда он установлен в '1' он указывает, что связанный с ним пакет имеет более высокий приоритет, чем другие пакеты, имеющие такой же PID, но у которых этот бит не установлен в '1'. Механизм транспортировки может использовать эту информацию для определения приоритетности передаваемых данных в рамках одного элементарного потока. В зависимости от приложения, поле transport_priority может кодироваться либо независимо от PID, либо в рамках только одного PID. Это поле может быть изменено канальными кодерами или декодерами.

PID – PID – это 13-битовое поле, указывающее тип данных, находящихся в полезной нагрузке пакета. Значение PID 0x0000 зарезервировано для Таблицы ассоциаций для программы (см. таблицу 2-30). Значение PID 0x0001 зарезервировано для Таблицы условного доступа (см. таблицу 2-32). Значение PID 0x0002 зарезервировано для Таблицы описания транспортного потока (см. таблицу 2-36), значение PID 0x0003 зарезервировано для Таблицы управляющей информации IPMP (см. ИСО/МЭК 13818-11) и значения PID 0x0004-0x000F зарезервированы. Значение PID 0x1FFF зарезервировано для нулевых пакетов (см. таблицу 2-3).

Таблица 2-3 – Таблица значений PID

Значение	Описание
0x0000	Таблица ассоциаций для программы
0x0001	Таблица условного доступа
0x0002	Таблица описания транспортного потока
0x0003	Таблица управляющей информации IPMP
0x0004-0x000F	Зарезервированы
0x0010 … 0x1FFE	Могут быть назначены в качестве network_PID, Program_map_PID, elementary_PID, или для других целей
0x1FFF	Нулевые пакеты
ПРИМЕЧАНИЕ. – Транспортные пакеты со значениями PID 0x0000, 0x0001 и 0x0010-0x1FFE могут содержать PCR.	

transport_scrambling_control – Это 2-битовое поле обозначает режим скремблирования полезной нагрузки пакета Транспортного потока. Заголовок пакета Транспортного потока и поле адаптации, когда они представлены, не должны подвергаться скремблированию. В случае нулевого пакета значение поля transport_scrambling_control должно быть установлено равным '00' (см. таблицу 2-4).

Таблица 2-4 – Значения управления скремблированием

Значение	Описание
00	Не скремблирован
01	Определяется пользователем
10	Определяется пользователем
11	Определяется пользователем

adaptation_field_control – Это 2-битовое поле указывает, следует ли после этого заголовка пакета Транспортного потока поле адаптации и/или полезная нагрузка (см. таблицу 2-5).

Таблица 2-5 – Значения управления полем адаптации

Значение	Описание
00	Зарезервировано ИСО/МЭК для будущего использования
01	Нет поля adaptation_field, только полезная нагрузка
10	Только поле adaptation_field, нет полезной нагрузки
11	Поле adaptation_field, за которым следует полезная нагрузка

Декодеры, соответствующие Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, должны отбрасывать пакеты Транспортного потока, в которых поле adaptation_field_control установлено в значение '00'. В случае нулевого пакета значение поля adaptation_field_control должно быть установлено равным '01'.

continuity_counter – Счетчик continuity_counter – это 4-битовое поле, увеличивающееся на единицу с прибытием каждого пакета Транспортного потока с одинаковым PID. После достижения максимального значения счетчик continuity_counter возвращается в 0. Счетчик continuity_counter не должен увеличиваться, когда значение adaptation_field_control в пакете равно '00' или '10'.

В Транспортных потоках двойные пакеты могут передаваться как два и только как два последовательных пакета Транспортного потока с одним PID. Дубликаты пакетов должны иметь то же значение счетчика continuity_counter, что и исходный пакет, а поле adaptation_field_control должно быть равно '01' или '11'. В дубликатах пакетов должен быть продублирован каждый байт исходного пакета, за исключением байтов, находящихся в полях эталонного времени программы, если они представлены, правильное значение должно быть закодировано.

Счетчик continuity_counter в конкретном пакете Транспортного потока является непрерывным, если он отличается на положительную величину "один" от значения continuity_counter в предыдущем пакете Транспортного потока с тем же самым PID, или, когда выполняется какое-либо из условий не увеличения значения счетчика (поле adaptation_field_control установлено в '00' или '10', или пакет является дубликатом, как описано выше). Непрерывность счетчика continuity_counter может быть нарушена, когда индикатор discontinuity_indicator установлен в '1' (см. § 2.4.3.4). Для нулевого пакета значение счетчика continuity_counter не определено.

data_byte – Байты данных должны быть непрерывными байтами данных пакетов PES (см. § 2.4.3.6), секциями PSI (см. § 2.4.4), байтами заполнения пакетов после секций PSI или конфиденциальной информацией, не передаваемой в этих структурах, как указано значением PID. В случае нулевых пакетов со значением PID = 0x1FFF, байтам data_byte может быть присвоено любое значение. Количество байтов data_byte (N) определяется как 184 минус количество байтов в поле adaptation_field(), как описано в § 2.4.3.4, ниже.

2.4.3.4 Поле адаптации

См. таблицу 2-6.

Таблица 2-6 – Поле адаптации Транспортного потока

Синтаксис	Количество битов	Мнемоника
adaptation_field() {		
adaptation_field_length	**8**	**uimsbf**
if (adaptation_field_length > 0) {		
discontinuity_indicator	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
elementary_stream_priority_indicator	**1**	**bslbf**
PCR_flag	**1**	**bslbf**
OPCR_flag	**1**	**bslbf**
splicing_point_flag	**1**	**bslbf**
transport_private_data_flag	**1**	**bslbf**
adaptation_field_extension_flag	**1**	**bslbf**
if (PCR_flag = = '1') {		
program_clock_reference_base	**33**	**uimsbf**
Reserved	**6**	**bslbf**
program_clock_reference_extension	**9**	**uimsbf**
}		
if (OPCR_flag = = '1') {		
original_program_clock_reference_base	**33**	**uimsbf**
Reserved	**6**	**bslbf**
original_program_clock_reference_extension	**9**	**uimsbf**
}		
if (splicing_point_flag = = '1') {		
splice_countdown	**8**	**tcimsbf**
}		
if (transport_private_data_flag = = '1') {		
transport_private_data_length	**8**	**uimsbf**
for (i = 0; i < transport_private_data_length; i++) {		
private_data_byte	**8**	**bslbf**
}		
}		
if (adaptation_field_extension_flag = = '1') {		
adaptation_field_extension_length	**8**	**uimsbf**
ltw_flag	**1**	**bslbf**
piecewise_rate_flag	**1**	**bslbf**
seamless_splice_flag	**1**	**bslbf**
Reserved	**5**	**bslbf**
if (ltw_flag = = '1') {		
ltw_valid_flag	**1**	**bslbf**
ltw_offset	**15**	**uimsbf**
}		
if (piecewise_rate_flag = = '1') {		
reserved	**2**	**bslbf**
piecewise_rate	**22**	**uimsbf**
}		
if (seamless_splice_flag = = '1') {		
Splice_type	**4**	**bslbf**
DTS_next_AU[32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
DTS_next_AU[29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
DTS_next_AU[14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
for (i = 0; i < N; i++) {		
reserved	**8**	**bslbf**
}		
}		
for (i = 0; i < N; i++) {		
stuffing_byte	**8**	**bslbf**
}		
}		
}		

2.4.3.5 Семантические определения полей в поле адаптации

adaptation_field_length – Поле adaptation_field_length – это 8-битовое поле, определяющее количество байтов в поле adaptation_field, которое следует сразу же после поля adaptation_field_length. Значение '0' означает введение в пакет Транспортного потока одного байта заполнения. Когда значение поля adaptation_field_control = '11', то значение поля adaptation_field_length должно находиться в диапазоне от 0 до 182. Когда значение поля adaptation_field_control = '10', то значение поля adaptation_field_length должно быть равно 183. Заполнение для пакетов Транспортного потока, в которых передаются пакеты PES, требуется, когда данных пакетов PES недостаточно для того, чтобы полностью заполнить байты полезной нагрузки пакета Транспортного потока. Заполнение выполняется путем определения более продолжительного поля адаптации, чем сумма длин входящих в него элементов данных, так что байты полезной нагрузки, остающиеся после поля адаптации, точно вмещают имеющиеся в наличии данные пакетов PES. Дополнительное пространство в поле адаптации заполняется байтами заполнения.

Для заполнения пакетов Транспортного потока, в которых передаются пакеты PES, может использоваться только один метод. Для пакетов Транспортного потока, в которых передаются данные PSI используется другой метод заполнения, который описан в § 2.4.4.

discontinuity_indicator – Это 1-битовое поле, которое, когда оно равно '1', указывает, что в текущем пакете Транспортного потока наблюдается нарушение непрерывности. Когда поле discontinuity_indicator равно '0', или оно не представлено, непрерывность не нарушена. Индикатор нарушения непрерывности используется для обозначения двух типов нарушения непрерывности – нарушения непрерывности временной базы системы и нарушения непрерывности счетчика continuity_counter.

Нарушение непрерывности временной базы системы обозначается путем использования в пакетах Транспортного потока индикатора discontinuity_indicator со значением PID, обозначенным как PCR_PID (см. § 2.4.4.9). Когда непрерывность для пакета Транспортного потока с PID, обозначенным как PCR_PID, нарушается, следующее значение эталонного времени программы (PCR) в пакете Транспортного потока с точно таким же значением PID представляет собой отсчет соответствующей программы с новым значением системных часов. Точка нарушения непрерывности временной базы системы определяется, как момент времени, в который на вход декодера T-STD поступает первый байт пакета, содержащего значение PCR новой временной базы системы. В том пакете, где возникает нарушение непрерывности временной базы системы, поле discontinuity_indicator должно быть установлено равным '1'. Бит discontinuity_indicator может быть также равен '1' в пакетах Транспортного потока с тем же значением PCR_PID, следующих до пакета, который содержит PCR новой временной базы системы. В таком случае, как только поле discontinuity_indicator устанавливается в '1', оно должно оставаться равным '1' во всех пакетах Транспортного потока с тем же значением PCR_PID вплоть до пакета Транспортного потока, содержащего первое значение PCR новой временной базы системы, включительно. После возникновения нарушения непрерывности временной базы системы, должно быть получено не менее двух PCR новой временной базы системы прежде, чем может возникнуть новое нарушение непрерывности временной базы системы. Далее, в наборе буферов декодера T-STD не может одновременно находиться более двух временных баз системы для одной программы, за исключением случая, когда состояние "Хитрого" режима = true.

Первый байт пакета Транспортного потока, содержащего метки времени PTS или DTS, обозначающие новую временную базу системы не должен поступать на вход декодера T-STD до нарушения непрерывности временной базы системы. После нарушения непрерывности временной базы системы, на вход декодера T-STD не должен поступать первый байт пакета Транспортного потока, содержащего метки времени PTS или DTS, обозначающие предыдущую временную базу системы.

Нарушение непрерывности счетчика continuity_counter обозначается путем использования индикатора discontinuity_indicator в любом пакете Транспортного потока. Когда непрерывность нарушается в любом пакете Транспортного потока с PID, не обозначенным как PCR_PID, значение поля continuity_counter в этом пакете может не быть непрерывным, относительно предшествующего пакета Транспортного потока с тем же самым PID. Когда непрерывность нарушается в пакете Транспортного потока со значением PID, обозначенным как PCR_PID, значение поля continuity_counter может не быть непрерывным только в пакете, в котором возникает нарушение непрерывности временной базы системы. Точка нарушения непрерывности счетчика непрерывности возникает, когда в пакете Транспортного потока состояние отсутствия непрерывности = true и поле continuity_counter в этом пакете не является непрерывным относительно предшествующего пакета Транспортного потока с тем же самым PID. Точка нарушения непрерывности счетчика непрерывности не должна появляться более одного раза с момента начала состояния отсутствия непрерывности и до выхода из состояния отсутствия непрерывности. Более того, для всех значений PID, которые не обозначены как PCR_PID, когда в пакете с определенным значением PID поле discontinuity_indicator = '1', поле discontinuity_indicator может быть равно '1' в следующем пакете Транспортного потока с тем же самым PID, но оно не должно быть равно '1' в трех последовательных пакетах Транспортного потока с одинаковым PID.

В данном пункте точка доступа элементарного потока определяется следующим образом:

- для видеоданных, соответствующих ИСО/МЭК 11172-2 и Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, – первый байт заголовка последовательности видеоданных;
- для видеоданных, соответствующих ИСО/МЭК 14496-2, – первый байт заголовка последовательности видеоданных;
- для видеоданных, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, – первый байт модуля доступа AVC. Набор параметров SPS и PPS, связанный с данным и всеми последующими модулями доступа AVC в кодированном аудиопотоке, должен быть установлен после данной точки доступа в потоке байтов и до их активации;
- для аудиоданных – первый байт звукового кадра.

После нарушения непрерывности счетчика в пакете Транспортного потока, который рассматривается как содержащий элементарный поток данных, первый байт этого потока в пакете Транспортного потока с тем же самым PID должен быть первым байтом точки доступа элементарного потока. В случае видеоданных, соответствующих ИСО/МЭК 11 172-2 или Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 14496-2, первый байт точки доступа элементарного потока может быть также первым байтом поля sequence_end_code, за которым следует точки доступа элементарного потока.

Каждый пакет Транспортного потока, который содержит данные элементарных потоков с PID, не обозначенным как PCR_PID, и в котором возникает нарушение непрерывности счетчика непрерывности, и в котором появляется метка PTS или DTS, должен поступать на вход декодера T-STD после появления нарушения непрерывности временной базы системы для соответствующей программы. Если в том случае, когда состояние отсутствия непрерывности = true, появляется два последовательных пакета Транспортного потока с одинаковым PID, которые имеют одно и то же значение поля continuity_counter и значения поля adaptation_field_control равные '01' или '11', второй пакет может быть отброшен. Транспортный поток не должен формироваться таким образом, чтобы отбрасывание этого пакета приводило бы к потере данных полезной нагрузки пакета PES или информации PSI.

После появления поля discontinuity_indicator, равного '1', в пакете Транспортного потока, который содержит информацию PSI, может появиться единичное нарушение непрерывности в секции version_number информации PSI. При возникновении такого нарушения непрерывности, номер версии должен быть передан в поле TS_program_map_sections соответствующей программы, в котором поле section_length = 13, и поле current_next_indicator = 1, так, чтобы не передавались никакие дескрипторы program_descriptor, и не описывались никакие элементарные потоки. После этого для каждой затронутой программы должна следовать секция TS_program_map_section, содержащая полное определение программы и имеющая увеличенный на единицу индикатор version_number ("номер версии"), и индикатор current_next_indicator = 1. Она указывает изменение версии в информации PSI.

random_access_indicator – Индикатор random_access_indicator – это 1-битовое поле, которое указывает, что текущий пакет Транспортного потока и, возможно, следующие пакеты Транспортного потока с тем же PID, содержат некоторую информацию, обеспечивающую возможность случайного доступа в этой точке.

Конкретно, если бит установлен в '1', то следующий пакет PES, с которого начинается полезная нагрузка пакетов Транспортного потока с текущим PID, должен содержать точку доступа элементарного потока, как определено в семантике поля discontinuity_indicator. Кроме того, в случае передачи видеоданных, в первом кадре, следующем за точкой доступа элементарного потока, должна присутствовать метка воспроизведения.

В случае передачи аудиоданных метка воспроизведения должна присутствовать в пакете PES, содержащем первый байт звукового кадра. В идентификаторе PCR_PID пакета Транспортного потока, содержащего поля PCR, индикатор random_access_indicator может быть установлен только в '1'.

elementary_stream_priority_indicator – Индикатор elementary_stream_priority_indicator – это 1-битовое поле. Оно указывает для пакетов с одним и тем же PID, приоритет данных в элементарных потоках, передаваемых в полезной нагрузке этого пакета Транспортного потока. Значение '1' указывает, что данная полезная нагрузка имеет более высокий приоритет, чем данные полезной нагрузки других пакетов Транспортного потока.

В случае передачи видеоданных, соответствующих ИСО/МЭК 11172-2, Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 14496-2, это поле может быть установлено в '1', только если полезная нагрузка содержит один или несколько байтов из секции внутреннего кодирования.

В случае передачи видеоданных, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, это поле может быть установлено в '1', только если полезная нагрузка содержит один или несколько байтов из секции со значениями поля slice_type, установленными в 2, 4, 7 или 9.

Значение '0' указывает, что полезная нагрузка имеет такой же приоритет, что и все остальные пакеты, в которых этот бит не установлен в '1'.

PCR_flag – Флаг PCR_flag – это 1-битовый флаг. Значение '1' указывает, что поле adaptation_field содержит поле PCR, закодированное в виде двух частей. Значение '0' указывает, что поле адаптации не содержит ни одного поля PCR.

OPCR_flag – Флаг OPCR_flag – это 1-битовый флаг. Значение '1' указывает, что поле adaptation_field содержит поле OPCR, закодированное в виде двух частей. Значение '0' указывает, что поле адаптации не содержит ни одного поля OPCR.

splicing_point_flag – Флаг splicing_point_flag – это 1-битовый флаг. Когда он установлен в '1', он указывает, что в соответствующем поле адаптации должно присутствовать поле splice_countdown, обозначая появление точки объединения. Значение '0' указывает, что в поле адаптации поле splice_countdown отсутствует.

transport_private_data_flag – Флаг transport_private_data_flag – это 1-битовый флаг. Значение '1' указывает, что поле адаптации содержит один или несколько байтов конфиденциальных данных (private_data). Значение '0' указывает, что поле адаптации не содержит ни одного байта private_data.

adaptation_field_extension_flag – Флаг adaptation_field_extension_flag – это 1-битовое поле, которое, когда оно установлено в '1', указывает на присутствие расширения поля адаптации. Значение '0' указывает, что расширение поля адаптации в поле адаптации отсутствует.

program_clock_reference_base; program_clock_reference_extension – Поле program_clock_reference (PCR) – это 42-битовое поле, закодированное в виде двух частей. Первая часть (program_clock_reference_base) – это 33-битовое поле, значение которого определяется значением PCR_base(i), как показано в уравнении 2-2. Вторая часть (program_clock_reference_extension) – это 9-битовое поле, значение которого определяется значением PCR_ext(i), как показано в уравнении 2-3. Поле PCR указывает предполагаемое время прибытия байта, содержащего последний бит поля program_clock_reference_base, на вход декодера конечной системы.

original_program_clock_reference_base; original_program_clock_reference_extension – Дополнительное поле эталонного времени исходной программы (OPCR) – это 42-битовое поле, закодированное в виде двух частей. Эти две части – база и расширение – кодируются так же, как и две соответствующие части поля PCR. Присутствие поля OPCR обозначается флагом OPCR_flag. Поле OPCR должно кодироваться только в тех пакетах Транспортного потока, в которых присутствует поле PCR. Поля OPCR разрешены как в однопрограммных, так и в многопрограммных Транспортных потоках.

Поле OPCR упрощает восстановление однопрограммного Транспортного потока из другого Транспортного потока. При восстановлении исходного однопрограммного Транспортного потока, поле OPCR может быть скопировано в поле PCR. Результирующее значение PCR действительно только, если исходный однопрограммный Транспортный поток восстанавливается в точности и во всей своей полноте. Оно будет включать в себя, как минимум, любые пакеты информации PSI и конфиденциальной информации, которые присутствовали в исходном Транспортном потоке и которые, возможно, потребуют других механизмов конфиденциальности. Это означает также, что поле OPCR должно быть точной копией соответствующего ему поля PCR в исходном однопрограммном Транспортном потоке.

Поле OPCR выражается следующим образом:

$$OPCR(i) = OPCR_base(i) \times 300 + OPCR_ext(i), \tag{2-8}$$

где:

$$OPCR_base(i) = ((system_clock_frequency \times t(i))\, DIV\, 300)\%2^{33} \tag{2-9}$$

$$OPCR_ext(i) = ((system_clock_frequency \times t(i))\, DIV\, 1)\%\ 300 \tag{2-10}$$

Декодер игнорирует поле OPCR. Ни декодер, ни мультиплексор не должны изменять поле OPCR.

splice_countdown – Поле splice_countdown – это 8-битовое поле, представляющее собой значение, которое может быть положительным или отрицательным. Положительное значение указывает оставшееся число пакетов Транспортного потока, с одинаковым PID, которые следуют после соответствующего пакета Транспортного потока до тех пор, пока не будет достигнута точка объединения. Дубликаты пакетов Транспортного потока и пакеты Транспортного потока, которые содержат только поля адаптации, исключаются. Точка объединения располагается непосредственно после последнего байта пакета Транспортного потока, в котором соответствующее поле splice_countdown достигает нулевого значения. В пакете Транспортного потока, в котором поле splice_countdown достигает нулевого значения, последний байт данных полезной нагрузки пакета Транспортного потока должен быть последним байтом кодированного звукового кадра или кодированного видеокадра. В случае видеопотока, соответствующий модуль доступа может завершаться или не завершаться кодом sequence_end_code. Пакеты Транспортного потока с одним и тем же PID, которые следуют далее, могут содержать данные другого элементарного потока того же типа.

Полезная нагрузка следующего пакета Транспортного потока с таким тем же PID (дубликаты пакетов и пакеты без полезной нагрузки исключаются) должна начинаться с первого байта пакета PES. В случае аудиопотока полезная нагрузка пакета PES должна начинаться с точки доступа. В случае видеопотока полезная нагрузка пакета PES должна начинаться с точки доступа, или с кода sequence_end_code, после которого следует точка доступа. Таким образом, предыдущий кодированный аудиокадр или кодированный видеокадр выравнивается по границе пакета, или заполняется с той же целью. Поле обратного счета может присутствовать также и после точки объединения. Когда поле обратного счета splice_countdown имеет отрицательное значение "минус n" (–n), оно указывает, что соответствующий пакет Транспортного потока является n-ным пакетом после точки объединения (дубликаты пакетов и пакеты без полезной нагрузки исключаются).

Точка доступа элементарного потока определена в семантике поля discontinuity_indicator.

transport_private_data_length – Поле transport_private_data_length – это 8-битовое поле, определяющее количество байтов в поле private_data, которое следует сразу же после поля transport_private_data_length. Количество байтов в поле private_data не должно быть таким, чтобы конфиденциальная информация выходила за пределы поля адаптации.

private_data_byte – Поле private_data_byte – это 8-битовое поле, которое в документах МСЭ-Т | ИСО/МЭК определяться не должно.

adaptation_field_extension_length – Поле adaptation_field_extension_length – это 8-битовое поле. Оно указывает количество байтов данных расширения поля адаптации, которое следует сразу же после этого поля, включая зарезервированные байты, если они присутствуют.

ltw_flag (legal time window_flag) – Это 1-битовое поле, которое, когда установлено в '1', указывает присутствие поля ltw_offset.

piecewise_rate_flag – Это 1-битовое поле, которое, когда установлено в '1', указывает присутствие поля piecewise_rate.

seamless_splice_flag – Это 1-битовый флаг, который, когда установлен в '1', указывает присутствие полей splice_type и DTS_next_AU. Значение '0' указывает, что нет ни поля splice_type, ни поля DTS_next_AU. Это поле не должно быть установлено в '1' в тех пакетах Транспортного потока, в которых флаг splicing_point_flag не установлен в '1'. После того, как в пакете Транспортного потока, в котором поле splice_countdown имеет положительное значение, этот флаг установлен в '1', он должен быть установлен в '1' во всех последующих пакетах Транспортного потока с таким же PID, в которых флаг splicing_point_flag установлен в '1', до тех пор, пока не появится пакет, в котором значение счетчика splice_countdown достигает нуля (включая этот пакет).

Когда этот флаг установлен, и если элементарный поток, передаваемый с данным PID, не является видеопотоком, соответствующим Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, то поле splice_type устанавливается в '0000'. Если элементарный поток, передаваемый с данным PID, является видеопотоком, соответствующим Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, то он должен удовлетворять ограничениям, обозначенным в значении поля splice_type.

ltw_valid_flag (legal time window_valid_flag) – Это 1-битовое поле, которое, когда установлено в '1', указывает, что значение поля ltw_offset должно быть действующим. Значение '0' указывает, что значение в поле ltw_offset не определено.

ltw_offset (legal time window offset) – Это 15-битовое поле, значение которого определяется только, если флаг ltw_valid имеет значение '1'. Если это поле определено, то сдвиг временного окна измеряется в единицах по ($300/f_s$) секунд, где f_s – частота системных часов программы, которой принадлежит это значение PID, и соответствует:

$$сдвиг = t_1(i) - t(i)$$

$$ltw_offset = offset // 1,$$

где i – индекс первого байта этого пакета Транспортного потока, сдвиг – это величина, закодированная в этом поле, t(i) – время прибытия байта i на вход декодера T-STD, и $t_1(i)$ – верхний предел времени интервала времени, который называется Окном декретного времени, которое связано с этим пакетом Транспортного потока.

Окно декретного времени имеет такое свойство, что, если данный Транспортный поток доставляется на вход декодера T-STD, начиная с момента времени $t_1(i)$, т. е. в конце его Окна декретного времени, и все остальные пакеты Транспортного потока той же самой программы доставляются в конце их Окон декретного времени, то:

- Для видеоданных – Буфер MB_n для данного PID в декодере T-STD должен содержать менее 184 байтов этих элементарных потоков на момент времени, когда на его вход поступает первый байт полезной нагрузки этого пакета Транспортного потока, и не должно возникать никаких нарушений работы буфера в декодере T-STD.
- Для аудиоданных – Буфер B_n для данного PID в декодере T-STD должен содержать менее (BS_{dec} + 1) байтов данных элементарных потоков на момент времени, когда на его вход поступает первый байт этого пакета Транспортного потока, и не должно возникать никаких нарушений работы буфера в декодере T-STD.

В зависимости от различных факторов, включая размер буфера MB_n и скорость передачи данных между MB_n и EB_n, можно определить другое время $t_0(i)$, так, что, если этот пакет доставляется в любое время в пределах интервала [$t_0(i)$, $t_1(i)$], не будет никаких нарушений работы буфера в декодере T-STD. Этот интервал времени называется Окном декретного времени. Значение t_0 в настоящей Рекомендации | Международном стандарте не определяется.

Информация в этом поле предназначена для таких устройств, как ре-мультимлексоры, которым эта информация может быть необходима для восстановления состояния буферов MB_n.

piecewise_rate – Значение этого 22-битового поля определяется только, когда оба флага ltw_flag и ltw_valid_flag установлены в '1'. Когда оно определено, оно представляет собой положительное целое число, определяющее гипотетическую скорость передачи R, которая используется для определения значений времени завершения Окон декретного времени пакетов Транспортного потока с одним и тем же PID, которые следуют после этого пакета, но не содержат поля legal_time_window_offset.

Предположим, что первый байт этого пакета Транспортного потока и N следующих пакетов Транспортного потока с одним и тем же PID имеют индексы A_i, A_{i+1}, ..., A_{i+N}, соответственно, и, что N последних пакетов не имеют значения, закодированного в поле legal_time_window_offset. Тогда значения $t_1(A_{i+j})$ должны определяться из выражения:

$$t_1(A_{1+j}) = t_1(A_i) + j \times 188 \times 8 \; \textit{битов / байт} / R \; ,$$

где j меняется от 1 до N.

Все пакеты между этим пакетом и следующим пакетом с таким же PID, содержащим поле legal_time_window_offset, должны рассматриваться, как имеющие значение:

$$\textit{сдвиг} = t_1(A_i) - t(A_i)$$

соответствующее значению $t_1(.)$, вычисленному по вышеприведенной формуле, и закодированное в поле legal_time_window_offset. t(j) – это время поступления j-того байта на вход декодера T-STD.

Значение этого поля не определяется, когда оно присутствует в пакете Транспортного потока, не содержащем поля legal_time_window_offset.

splice_type – Это 4-битовое поле. С момента своего первого появления и на протяжении всего времени это поле должно иметь одно и то же значение во всех последующих пакетах Транспортного потока с одним и тем же PID, в которых оно присутствует, до появления пакета, в котором значение обратного счетчика splice_countdown достигает нуля (включая этот пакет). Если элементарный поток, передаваемый с данным PID, не является видеопотоком, соответствующим Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, то это поле имеет значение '0000'. Если это видеопоток, соответствующий Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, то поле показывает условия, которым должен удовлетворять данный элементарный поток в целях объединения. Эти условия определяются, как функция профиля, уровня и типа объединения (splice_type) в таблицах с 2-7 по 2-20.

В этих таблицах значение полей 'splice_decoding_delay' (задержка декодирования объединения) и 'max_splice_rate' (максимальная скорость передачи при объединении) означают, что для элементарного видеопотока должны выполняться следующие условия:

1) Последний байт кадра кодированного изображения, завершающегося в пакете Транспортного потока, в котором значение счетчика splice_countdown достигает нуля, должен оставаться в буфере VBV модели VBV в течение интервала времени, равного (splice_decoding_delay $t_{n+1} - t_n$), где, для настоящего подраздела, используются следующие обозначения:
   - n – индекс кадра кодированного изображения, завершающегося в пакете Транспортного потока, в котором значение счетчика splice_countdown достигает нуля, т. е. указанного выше кадра кодированного изображения.
   - t_n определяется в разделе C.3.1 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2.
   - ($t_{n+1} - t_n$) определяется в разделах C.9–C.12 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2.

ПРИМЕЧАНИЕ. – t_n – это момент времени, в который n-ый кадр кодированного изображения удаляется из буфера VBV, и (t_{n+1} – t_n) – продолжительность времени, в течение которого воспроизводится n-ый кадр изображения.

2) Буфер VBV модели VBV не должен переполняться. Если его вход подключен к точке объединения для потока с постоянной скоростью передачи, равной 'max_splice_rate', в течение времени, равного 'splice_decoding_delay'.

Таблица 2-7 – Таблица параметров объединения №1 – Видеоданные
Основной уровень с простым профилем, Основной уровень с основным профилем, Основной уровень с профилем SNR (оба уровня), Уровень Высокий-1440 с пространственным профилем (базовый уровень), Основной уровень с высоким профилем (средний и базовый уровни), Основной уровень с многопроекционным профилем (базовый уровень)

splice_type	Условия
0000	splice_decoding_delay = 120 мс; max_splice_rate = 15.0 × 10^6 бит/с
0001	splice_decoding_delay = 150 мс; max_splice_rate = 12.0 × 10^6 бит/с
0010	splice_decoding_delay = 225 мс; max_splice_rate = 8.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 7.2 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-8 – Таблица параметров объединения №2 – Видеоданные
Низкий уровень с основным профилем, Низкий уровень с профилем SNR (оба уровня), Основной уровень с высоким профилем (базовый уровень), Низкий уровень с многопроекционным профилем (базовый уровень)

splice_type	Условия
0000	splice_decoding_delay = 115 мс; max_splice_rate = 4.0 × 10^6 бит/с
0001	splice_decoding_delay = 155 мс; max_splice_rate = 3.0 × 10^6 бит/с
0010	splice_decoding_delay = 230 мс; max_splice_rate = 2.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 1.8 ×10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-9 – Таблица параметров объединения №3 – Видеоданные
Уровень Высокий-1440 с основным профилем, Уровень Высокий-1440 с пространственным профилем (все уровни), Уровень Высокий-1440 с высоким профилем (средний и базовый уровни), Уровень Высокий-1440 с многопроекционным профилем (базовый уровень)

splice_type	Условия
0000	splice_decoding_delay = 120 мс; max_splice_rate = 60.0 × 10^6 бит/с
0001	splice_decoding_delay = 160 мс; max_splice_rate = 45.0 × 10^6бит/с
0010	splice_decoding_delay = 240 мс; max_splice_rate = 30.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 28.5 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-10 – Таблица параметров объединения №4 – Видеоданные
Высокий уровень с основным профилем, Уровень Высокий-1440 с высоким профилем (все уровни), Высокий уровень с высоким профилем (средний и базовый уровни), Высокий уровень с многопроекционным профилем (базовый уровень)

splice_type	Условия
0000	splice_decoding_delay = 120 мс; max_splice_rate = 80.0 × 10^6 бит/с
0001	splice_decoding_delay = 160 мс; max_splice_rate = 60.0 × 10^6 бит/с
0010	splice_decoding_delay = 240 мс; max_splice_rate = 40.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 38.0 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-11 – Таблица параметров объединения №5 – Видеоданные
Низкий уровень с профилем SNR (базовый уровень)

splice_type	Условия
0000	splice_decoding_delay = 115 мс; max_splice_rate = 3.0 × 10^6 бит/с
0001	splice_decoding_delay = 175 мс; max_splice_rate = 2.0 × 10^6 бит/с
0010	splice_decoding_delay = 250 мс; max_splice_rate = 1.4 × 10^6 бит/с
0011-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-12 – Таблица параметров объединения №6 – Видеоданные
Основной уровень с профилем SNR (базовый уровень)

splice_type	Условия
0000	splice_decoding_delay = 115 мс; max_splice_rate = 10.0 × 10^6 бит/с
0001	splice_decoding_delay = 145 мс; max_splice_rate = 8.0 × 10^6 бит/с
0010	splice_decoding_delay = 235 мс; max_splice_rate = 5.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 4.7 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-13 – Таблица параметров объединения №7 – Видеоданные
Уровень Высокий-1440 с пространственным профилем (средний и базовый уровни)

splice_type	Условия
0000	splice_decoding_delay = 120 мс; max_splice_rate = 40.0 × 10^6 бит/с
0001	splice_decoding_delay = 160 мс; max_splice_rate = 30.0 × 10^6 бит/с
0010	splice_decoding_delay = 240 мс; max_splice_rate = 20.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 19.0 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-14 – Таблица параметров объединения №8 – Видеоданные Основной уровень с высоким профилем (все уровни), Уровень Высокий-1440 с высоким профилем (базовый уровень)

splice_type	Условия
0000	splice_decoding_delay = 120 мс; max_splice_rate = 20.0 × 10^6 бит/с
0001	splice_decoding_delay = 160 мс; max_splice_rate = 15.0 × 10^6 бит/с
0010	splice_decoding_delay = 240 мс; max_splice_rate = 10.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 9.5 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-15 – Таблица параметров объединения №9 – Видеоданные Высокий уровень с высоким профилем (базовый уровень), Основной уровень с многопроекционным профилем (оба уровня)

splice_type	Условия
0000	splice_decoding_delay = 120 мс; max_splice_rate = 25.0 × 10^6 бит/с
0001	splice_decoding_delay = 165 мс; max_splice_rate = 18.0 × 10^6 бит/с
0010	splice_decoding_delay = 250 мс; max_splice_rate = 12.0 × 10^6 бит/с
0011-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-16 – Таблица параметров объединения №10 – Видеоданные Высокий уровень с высоким профилем (все уровни), Уровень Высокий-1440 с многопроекционным профилем (оба уровня)

splice_type	Условия
0000	splice_decoding_delay = 120 мс; max_splice_rate = 100.0 × 10^6 бит/с
0001	splice_decoding_delay = 160 мс; max_splice_rate = 75.0 × 10^6 бит/с
0010	splice_decoding_delay = 240 мс; max_splice_rate = 50.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 48.0 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-17 – Таблица параметров объединения №11 – Видеоданные Основной уровень с профилем 4:2:2

splice_type	Условия
0000	splice_decoding_delay = 45 мс; max_splice_rate = 50.0 × 10^6 бит/с
0001	splice_decoding_delay = 90 мс; max_splice_rate = 50.0 × 10^6 бит/с
0010	splice_decoding_delay = 180 мс; max_splice_rate = 50.0 × 10^6 бит/с
0011	splice_decoding_delay = 225 мс; max_splice_rate = 40.0 × 10^6 бит/с
0100	splice_decoding_delay = 250 мс; max_splice_rate = 36.0 × 10^6 бит/с
0101-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-18 – Таблица параметров объединения №12 – Видеоданные Низкий уровень с многопроекционным профилем (оба уровня)

splice_type	Условия
0000	splice_decoding_delay = 115 мс; max_splice_rate = 8.0 × 10^6 бит/с
0001	splice_decoding_delay = 155 мс; max_splice_rate = 6.0 × 10^6 бит/с
0010	splice_decoding_delay = 230 мс; max_splice_rate = 4.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 3.7 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-19 – Таблица параметров объединения №13 – Видеоданные Высокий уровень с многопроекционным профилем (оба уровня)

splice_type	Условия
0000	splice_decoding_delay = 120 мс; max_splice_rate = 130.0 × 10^6 бит/с
0001	splice_decoding_delay = 150 мс; max_splice_rate = 104.0 × 10^6 бит/с
0010	splice_decoding_delay = 240 мс; max_splice_rate = 65.0 × 10^6 бит/с
0011	splice_decoding_delay = 250 мс; max_splice_rate = 62.4 × 10^6 бит/с
0100-1011	Зарезервированы
1100-1111	Определяется пользователем

Таблица 2-20 – Таблица параметров объединения №14 – Видеоданные Высокий уровень с профилем 4:2:2

splice_type	Условия
0000	splice_decoding_delay = 45 мс; max_splice_rate = 300.0 × 10^6 бит/с
0001	splice_decoding_delay = 90 мс; max_splice_rate = 300.0 × 10^6 бит/с
0010-0011	Зарезервированы
0100	splice_decoding_delay = 250 мс; max_splice_rate = 180.0 × 10^6 бит/с
0101-1011	Зарезервированы
1100-1111	Определяется пользователем

DTS_next_AU (метка времени декодирования следующего модуля) – Это 33-битовое поле, закодированное в виде трех частей. В случае непрерывного и периодического декодирования в этой точке объединения, оно указывает время декодирования первого модуля доступа после точки объединения. Это время декодирования выражается в единицах временной базы, которая действует в том пакете Транспортного потока, в котором достигает нуля значение счетчика splice_countdown. После первого своего появления это поле должно иметь одно и то же значение во всех следующих пакетах Транспортного потока с одинаковым PID, в которых оно присутствует, до появления пакета, в котором значение счетчика splice_countdown достигает нуля (включая этот пакет).

stuffing_byte – Это фиксированное 8-битовое значение, равное '1111 1111', которое может быть введено кодером. Оно отбрасывается декодером.

2.4.3.6 Пакет PES

См. таблицу 2-21.

Таблица 2-21 – Пакет PES

Синтаксис	Количество битов	Мнемоника
PES_packet() {		
packet_start_code_prefix	**24**	**bslbf**
stream_id	**8**	**uimsbf**
PES_packet_length	**16**	**uimsbf**
if (stream_id != program_stream_map		
&& stream_id != padding_stream		
&& stream_id != private_stream_2		
&& stream_id != ECM		
&& stream_id != EMM		
&& stream_id != program_stream_directory		
&& stream_id != DSMCC_stream		
&& stream_id != ITU-T Rec. H.222.1 type E stream) {		
'10'	**2**	**bslbf**
PES_scrambling_control	**2**	**bslbf**
PES_priority	**1**	**bslbf**
data_alignment_indicator	**1**	**bslbf**
copyright	**1**	**bslbf**
original_or_copy	**1**	**bslbf**
PTS_DTS_flags	**2**	**bslbf**
ESCR_flag	**1**	**bslbf**
ES_rate_flag	**1**	**bslbf**
DSM_trick_mode_flag	**1**	**bslbf**
additional_copy_info_flag	**1**	**bslbf**
PES_CRC_flag	**1**	**bslbf**
PES_extension_flag	**1**	**bslbf**
PES_header_data_length	**8**	**uimsbf**
if (PTS_DTS_flags == '10') {		
'0010'	**4**	**bslbf**
PTS [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
if (PTS_DTS_flags == '11') {		
'0011'	**4**	**bslbf**
PTS [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
'0001'	**4**	**bslbf**
DTS [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
DTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
DTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
if (ESCR_flag == '1') {		
Зарезервированы	**2**	**bslbf**
ESCR_base[32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
ESCR_base[29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
ESCR_base[14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
ESCR_extension	**9**	**uimsbf**
marker_bit	**1**	**bslbf**
}		

Синтаксис	Количество битов	Мнемоника
if (ES_rate_flag == '1') {		
marker_bit	**1**	**bslbf**
ES_rate	**22**	**uimsbf**
marker_bit	**1**	**bslbf**
}		
if (DSM_trick_mode_flag == '1') {		
trick_mode_control	**3**	**uimsbf**
if (trick_mode_control == fast_forward) {		
field_id	**2**	**bslbf**
intra_slice_refresh	**1**	**bslbf**
frequency_truncation	**2**	**bslbf**
}		
else if (trick_mode_control == slow_motion) {		
rep_cntrl	**5**	**uimsbf**
}		
else if (trick_mode_control == freeze_frame) {		
field_id	**2**	**uimsbf**
Зарезервированы	**3**	**bslbf**
}		
else if (trick_mode_control == fast_reverse) {		
field_id	**2**	**bslbf**
intra_slice_refresh	**1**	**bslbf**
frequency_truncation	**2**	**bslbf**
else if (trick_mode_control == slow_reverse) {		
rep_cntrl	**5**	**uimsbf**
}		
Else		
Зарезервированы	**5**	**bslbf**
}		
if (additional_copy_info_flag == '1') {		
marker_bit	**1**	**bslbf**
additional_copy_info	**7**	**bslbf**
}		
if (PES_CRC_flag == '1') {		
previous_PES_packet_CRC	**16**	**bslbf**
}		
if (PES_extension_flag == '1') {		
PES_private_data_flag	**1**	**bslbf**
pack_header_field_flag	**1**	**bslbf**
program_packet_sequence_counter_flag	**1**	**bslbf**
P-STD_buffer_flag	**1**	**bslbf**
Зарезервированы	**3**	**bslbf**
PES_extension_flag_2	**1**	**bslbf**
if (PES_private_data_flag == '1') {		
PES_private_data	**128**	**bslbf**
}		
if (pack_header_field_flag == '1') {		
pack_field_length	**8**	**uimsbf**
pack_header()		
}		
if (program_packet_sequence_counter_flag == '1') {		
marker_bit	**1**	**bslbf**
program_packet_sequence_counter	**7**	**uimsbf**
marker_bit	**1**	**bslbf**
MPEG1_MPEG2_identifier	**1**	**bslbf**
original_stuff_length	**6**	**uimsbf**
}		
if (P-STD_buffer_flag == '1') {		
'01'	**2**	**bslbf**
P-STD_buffer_scale	**1**	**bslbf**
P-STD_buffer_size	**13**	**uimsbf**
}		
if (PES_extension_flag_2 == '1') {		
marker_bit	**1**	**bslbf**
PES_extension_field_length	**7**	**uimsbf**
stream_id_extension_flag	**1**	**bslbf**
If (stream_id_extension_flag == '0') {		

Синтаксис	Количество битов	Мнемоника
stream_id_extension	**7**	**uimsbf**
for (i = 0; i < PES_extension_field_length; i++){		
reserved	**8**	**bslbf**
}		
}		
}		
}		
for (i < 0; i < N1; i++) {		
stuffing_byte	**8**	**bslbf**
}		
for (i < 0; i < N2; i++) {		
PES_packet_data_byte	**8**	**bslbf**
}		
}		
else if (stream_id == program_stream_map		
|| stream_id == private_stream_2		
|| stream_id == ECM		
|| stream_id == EMM		
|| stream_id == program_stream_directory		
|| stream_id == DSMCC_stream		
|| stream_id == ITU-T Rec. H.222.1 type E stream) {		
for (i = 0; i < PES_packet_length; i++) {		
PES_packet_data_byte	**8**	**bslbf**
}		
}		
else if (stream_id == padding_stream) {		
for (i < 0; i < PES_packet_length; i++) {		
padding_byte	**8**	**bslbf**
}		
}		
}		

2.4.3.7 Семантические определения полей в пакете PES

packet_start_code_prefix – Поле packet_start_code_prefix – это 24-битовый код. Вместе со следующим после него идентификатором потока (stream_id), он образует стартовый код пакета, который обозначает начало пакета. Поле packet_start_code_prefix – это строка битов '0000 0000 0000 0000 0000 0001' (0x000001).

stream_id – В Программных потоках поле stream_id указывает тип и номер элементарного потока, определяемого значением stream_id в таблице 2-22. В Транспортных потоках поле stream_id может быть установлено в любое действующее значение, которое правильно описывает тип элементарного потока, как определено в таблице 2-22. В Транспортных потоках тип элементарного потока определяется в Специальной информации программы, которая определена в § 2.4.4.

PES_packet_length –16-битовое поле, определяющее количество байтов в пакете PES, которые следуют после последнего байта этого поля. Значение 0 указывает, что длина пакета PES не определена и не ограничена. Оно может присутствовать только в тех пакетах PES, полезная нагрузка которых состоит из байтов элементарного видеопотока, содержащихся в пакетах Транспортного потока.

PES_scrambling_control –2-битовое поле PES_scrambling_control указывает режим скремблирования полезной нагрузки пакета PES. Когда скремблирование выполняется на уровне PES, заголовок пакета PES, включая дополнительные поля, когда они присутствуют, не должен подвергаться скремблированию (см. таблицу 2-23).

Таблица 2-22 – Присвоение идентификатора потока Stream_id

Stream_id	Примечание	Кодировка потока
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		номер аудиопотока x xxxx согласно ИСО/МЭК 13818-3 или ИСО/МЭК 11172-3 или ИСО/МЭК 13818-7 или ИСО/МЭК 14496-3
1110 xxxx		номер видеопотока xxxx согласно Рек. МСЭ-Т H.262 | ИСО/МЭК 13818-2, ИСО/МЭК 11172-2, ИСО/МЭК 14496-2 или Рек. МСЭ-Т H.264 | ИСО/МЭК 14496-10

Таблица 2-22 – Присвоение идентификатора потока Stream_id

Stream_id	Примечание	Кодировка потока
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	Рекомендация МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 Приложение A или ИСО/МЭК 13818-6_DSMCC_stream
1111 0011	2	ИСО/МЭК_13522_stream
1111 0100	6	Рекомендация. МСЭ-Т H.222.1 тип A
1111 0101	6	Рекомендация МСЭ-Т H.222.1 тип B
1111 0110	6	Рекомендация МСЭ-Т H.222.1 тип C
1111 0111	6	Рекомендация МСЭ-Т H.222.1 тип D
1111 1000	6	Рекомендация МСЭ-Т H.222.1 тип E
1111 1001	7	ancillary_stream
1111 1010		ИСО/МЭК 14496-1_SL-packetized_stream
1111 1011		ИСО/МЭК 14496-1_FlexMux_stream
1111 1100		Поток метаданных
1111 1101	8	extended_stream_id
1111 1110		Резервный поток данных
1111 1111	4	program_stream_directory

x означает, что разрешены значения и '0', и '1', и их использование приводит к тому же типу потока. Номер потока задается значениями, которые принимает x.

ПРИМЕЧАНИЕ 1. – Пакеты PES типа program_stream_directory имеют уникальный синтаксис, описанный в § 2.5.4.1.

ПРИМЕЧАНИЕ 2. – Пакеты PES типа private_stream_1 и ISO/IEC_13552_stream имеют такой же синтаксис пакетов PES, что и в видеопотоках, соответствующих Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, и аудиопотоках, соответствующих стандарту ИСО/МЭК 13818-3.

ПРИМЕЧАНИЕ 3. – Пакеты PES типа private_stream_2, ECM_stream и EMM_stream аналогичны пакетам типа private_stream_1, за исключением того, что синтаксис после поля PES_packet_length в них не определен.

ПРИМЕЧАНИЕ 4. – Пакеты PES типа program_stream_directory имеют уникальный синтаксис, описанный в § 2.5.5.

ПРИМЕЧАНИЕ 5. – Пакеты PES типа DSM-CC_stream имеют уникальный синтаксис, определенный в стандарте ИСО/МЭК 13818-6.

ПРИМЕЧАНИЕ 6. – Этот идентификатор stream_id связан с полем stream_type 0x09 в таблице 2-34.

ПРИМЕЧАНИЕ 7. – Этот идентификатор stream_id используется только в пакетах PES, которые передают данные из программного или системного потока, соответствующего ИСО/МЭК 11172-1, в транспортный поток (см. § 2.4.3.8).

ПРИМЕЧАНИЕ 8. – Использование идентификатора stream_id 0xFD (extended_stream_id) показывает, что этот пакет PES использует расширенный синтаксис, чтобы разрешить идентификацию дополнительных типов потоков.

Таблица 2-23 – Управляющие значения для скремблирования PES

Значение	Описание
00	Не скремблируется
01	Определяется пользователем
10	Определяется пользователем
11	Определяется пользователем

PES_priority – Это 1-битовое поле, указывающее приоритет полезной нагрузки в этом пакете PES. '1' обозначает более высокий приоритет полезной нагрузки пакета PES, чем приоритет полезной нагрузки пакета PES, в котором это поле установлено в '0'. Мультиплексор может использовать бит PES_priority для определения приоритетности своих данных в элементарном потоке. Механизм транспортировки не должен менять это поле.

data_alignment_indicator – Это 1-битовый флаг. Когда он установлен в значение '1', то это означает, что сразу после заголовка пакета PES следует элемент видеосинтаксиса или слово звуковой синхронизации, обозначенные в поле data_stream_alignment_descriptor, описанном в § 2.6.10, если этот дескриптор присутствует. Если этот флаг установлен в '1', а дескриптор отсутствует, то требуется выравнивание, показанное в поле alignment_type '01' в таблице 2-53, таблице 2-54 или таблице 2-55. Если этот флаг установлен в '0', то не определено, происходит ли подобное выравнивание.

copyright – Это 1-битовое поле. Когда оно установлено в '1', оно указывает, что авторские права на материал полезной нагрузки соответствующего пакета PES защищены. Когда оно установлено в '0', то не определено, защищены ли авторские права на материал. Дескриптор авторского права, описанный в § 2.6.24, связан с элементарным потоком, который содержит этот Пакет PES, и флаг авторских прав установлен в '1', если этот дескриптор применяется к материалу, содержащемуся в этом пакете PES.

original_or_copy – Это 1-битовое поле. Когда оно установлено в '1', содержание полезной нагрузки связанного с ним пакета PES является оригиналом. Когда оно установлено в '0', это указывает, что содержание полезной нагрузки связанного с ним пакета PES является копией.

PTS_DTS_flags – Это 2-битовое поле. Когда поле PTS_DTS_flags установлено в '10', в заголовке пакета PES должны присутствовать поля PTS. Когда поле PTS_DTS_flags установлено в '11', в заголовке пакета PES должны присутствовать и поля PTS, и поля DTS. Когда поле PTS_DTS_flags установлено в '00' в заголовке пакета PES не должны присутствовать ни поля PTS, ни поля DTS. Значение '01' является запрещенным.

ESCR_flag – 1-битовый флаг, который, когда установлен в '1' указывает, что в заголовке пакета PES присутствует базовое поле ESCR и поля расширения. Когда он установлен в '0', он указывает, что полей ESCR нет.

ES_rate_flag – 1-битовый флаг, который, когда установлен в '1' указывает, что в заголовке пакета PES присутствует поле ES_rate. Когда он установлен в '0', он указывает, что поля ES_rate нет.

DSM_trick_mode_flag – 1-битовый флаг, который, когда установлен в '1' указывает присутствие 8-битового поля "хитрого" режима. Когда он установлен в '0', он указывает, что этого поля нет.

additional_copy_info_flag – 1-битовый флаг, который, когда установлен в '1', он указывает наличие дополнительного поля additional_copy_info. Когда он установлен в '0', он указывает, что этого поля нет.

PES_CRC_flag – 1-битовый флаг, который, когда установлен в '1' указывает, что в пакете PES присутствует поле CRC. Когда он установлен в '0', он указывает, что этого поля нет.

PES_extension_flag – 1-битовый флаг, который, когда установлен в '1' указывает, что в заголовке этого пакета PES присутствует поле расширения. Когда он установлен в '0', он указывает, что этого поля нет.

PES_header_data_length – 8-битовое поле, указывающее общее число байтов, занимаемых дополнительными полями, и любых байтов заполнения, содержащихся в заголовке этого пакета PES. Присутствие дополнительных полей указывается байтом, который предшествует полю PES_header_data_length.

marker_bit – marker_bit– это 1-битовое поле, которое имеет значение '1'.

PTS (метка времени воспроизведения) – Моменты времени воспроизведения должны быть связаны с моментами времени декодирования следующим образом: Метка PTS – это 33-битовое число, закодированное в трех отдельных полях. Оно указывает время воспроизведения $tp_n(k)$ в декодере конечной системы модуля представления k из элементарного потока n. Значение метки PTS определяется в единицах периода системных часов, разделенного на 300 (что дает в результате 90 кГц). Время воспроизведения вычисляется из метки PTS в соответствии с приведенным ниже уравнением 2-11. Ограничения по частоте для кодирования меток времени воспроизведения описаны в § 2.7.4.

$$PTS(k) = ((system_clock_frequency \times tp_n(k))DIV\ 300)\%\ 2^{33}, \qquad (2\text{-}11)$$

где $tp_n(k)$ – время воспроизведения модуля представления $P_n(k)$.

В случае аудиопотока, если в заголовке пакета PES присутствует метка PTS, она должна указывать на первый модуль доступа, начинающийся в пакете PES. Звуковой модуль доступа начинается в пакете PES, если в пакете PES присутствует первый байт звукового модуля доступа.

В случае видеопотока, соответствующего ИСО/МЭК 11172-2, Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 14496-2, если в заголовке пакета PES присутствует метка PTS, то она должна ссылаться на модуль доступа, содержащий стартовый код первого кадра изображения, начинающегося в данном пакете PES. Стартовый код кадра изображения начинается в пакете PES, если в нем присутствует первый байт стартового кода. Для I- и P-изображений в последовательностях не с малой задержкой (non-low_delay) и в случае, если между модулями доступа (AU) k и k' нет нарушений непрерывности декодирования, время воспроизведения $tp_n(k)$ должно быть равно времени декодирования $td_n(k')$ следующего передаваемого I- или P-изображения (см. § 2.7.5). Если имеет место нарушение непрерывности декодирования, или если поток заканчивается, то разница между $tp_n(k)$ и $td_n(k)$ будет той же самой, как если бы первоначальный поток продолжался без нарушений непрерывности и не заканчивался.

ПРИМЕЧАНИЕ 1 – Последовательность с малой задержкой – это видеопоследовательность, соответствующая ИСО/МЭК 14496-2, в которой флаг малой задержки (low_delay) установлен в '1' (см. § 6.2.3 стандарта ИСО/МЭК 14496-2).

В случае видеопотока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, если в заголовке пакета PES присутствует метка PTS, она должна ссылаться на модуль доступа, содержащий стартовый код первого кадра изображения, который начинается в данном пакете PES. Стартовый код кадра изображения начинается в пакете PES, если в данном в пакете PES присутствует первый байт стартового кода. Для кадров с I- и P-кодировкой в последовательностях не с малой задержкой (non-low_delay) и в случае, если между модулями доступа (AU) k и k' нет нарушений непрерывности декодирования, время воспроизведения $tp_n(k)$ должно быть равно времени декодирования $t_{dn}(k')$ следующего передаваемого I- или P-изображения (см. § 2.7.5). Если имеет место нарушение непрерывности декодирования, или если поток заканчивается, то разница между $t_{pn}(k)$ и $t_{dn}(k)$ должна быть той же самой, как если бы первоначальный поток продолжался без нарушений непрерывности и не заканчивался.

ПРИМЕЧАНИЕ 2. – Последовательность с малой задержкой – это видеопоследовательность, соответствующая ИСО/МЭК 14496-2, в которой флаг малой задержки (low_delay) установлен в '1' (см. § 6.2.3 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2). Отметим также, что для кадров с чересстрочной разверткой время воспроизведения определяется первым полем изображения кодированного кадра.

В случае видеопотока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, если в заголовке пакета PES присутствует метка PTS, она должна ссылаться на первый модуль доступа AVC, который начинается в данном пакете PES. Модуль доступа AVC начинается в пакете PES, если в данном в пакете PES присутствует первый байт модуля доступа AVC. Для обеспечения согласованности между моделью STD и моделью гипотетического эталонного декодера (hypothetical reference decoder (HRD)), определенной в Приложении C Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, для каждого декодированного модуля доступа AVC, значение метки PTS в декодере STD должно, с точностью соответствующих часов, указывать тот же самый момент времени, что и номинальное время выходного сигнала DPB в HRD, которое определено здесь как $t_{o,n,dpb}(n) = t_{r,n}(n) + t_c * dpb_output_delay(n)$, где $t_{r,n}(n)$, t_c и dpb_output_delay(n) определяются в соответствии с Приложением C Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10.

ПРИМЕЧАНИЕ 3. – Для вычисления метки PTS и времени $t_{o,n,dpb}(n)$ могут быть использованы различные часы.

Время воспроизведения $t_{pn}(k)$ должно быть равно времени декодирования $t_{dn}(k)$ для:

- звуковых модулей доступа;
- модулей доступа в последовательностях с малой задержкой, соответствующих Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 14496-2;
- B-кадров изображения в видеопотоках, соответствующих ИСО/МЭК 11172-2, Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 14496-2.

Если к аудиоданным применяется фильтрация, то в системной модели предполагается, что фильтрация не вносит никаких задержек, следовательно, отсчет, обозначенный в процессе кодирования меткой PTS, является при декодировании тем же самым отсчетом, обозначенным меткой PTS. Случай масштабирующего кодирования описывается в § 2.7.6.

DTS (метка времени декодирования) – это 33-битовое число, закодированное в трех отдельных полях. Оно указывает время декодирования $td_n(j)$ модуля доступа j элементарного потока n в декодере конечной системы. Значение метки DTS определяется в единицах периода системных часов, разделенного на 300 (что дает в результате 90 кГц). Время декодирования вычисляется из DTS в соответствии с приведенным ниже уравнением 2-12:

$$DTS(j) = ((system_clock_frequency \times td_n(j))DIV\ 300)\%\ 2^{33}, \qquad (2\text{-}12)$$

где $td_n(j)$ – время декодирования модуля доступа $A_n(j)$.

В случае видеопотока, соответствующего ИСО/МЭК 11172-2, Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 14496-2, если в заголовке пакета PES присутствует метка DTS, он должен указывать на модуль доступа, содержащий начальный код первого изображения, начинающийся в данном пакете PES. Стартовый код изображения начинается в пакете PES, если в этом пакете PES присутствует его первый байт.

Для видеопотока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, если в заголовке пакета PES присутствует метка DTS, она должна указывать на первый модуль доступа AVC, который начинается в данном пакете PES. Модуль доступа AVC начинается в пакете PES, если в пакете присутствует его первый байт. Для того чтобы достичь соответствия между моделью STD и моделью HRD, определенной в Приложении C Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, в декодере STD значение метки DTS для каждого модуля доступа AVC должно – в пределах точности часов – одно и то же мгновенное значение номинального времени выходного сигнала CPB $t_{r,n}(n)$ в HRD, как определено в Приложении C Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10.

ПРИМЕЧАНИЕ 4. – Для вычисления отклонения DTS и времени $t_{r,n}(n)$ могут использоваться разные часы.

Случай масштабируемого кодирования описывается в § 2.7.6.

ESCR_base; ESCR_extension – это 42-битовое поле, кодированное в виде двух частей. Первая часть – ESCR_base – это 33-битовое поле, значение которого определяется полем ESCR_base(i), как показано в уравнении 2-14. Вторая часть – ESCR_ext – это 9-битовое поле, значение которого определяется полем ESCR_ext(i), как показано в уравнении 2-15. Поле ESCR указывает планируемое время прибытия байта, содержащего последний бит поля ESCR_base, на вход декодера PES-STD для потоков PES (см. § 2.5.2.4).

В частности:

$$ESCR(i) = ESCR_base(i) \times 300 + ESCR_ext(i)\,, \tag{2-13}$$

где:

$$ESCR_base(i) = ((system_clock_frequency \times t(i))DIV\ 300)\%\ 2^{33} \tag{2-14}$$

$$ESCR_ext(i) = ((system_clock_frequency \times t(i))DIV\ 1)\%\ 300 \tag{2-15}$$

Поле ESCR и поле ES_rate (его семантика описана ниже сразу после настоящего параграфа) содержат информацию синхронизации, относящуюся к последовательности потоков PES. Для этих полей должны соблюдаться ограничения, определенные в § 2.7.3.

ES_rate (скорость передачи элементарного потока) – Поле ES_rate представляет собой 22-битовое целое число без знака, указывающее скорость, с которой декодер конечной системы принимает байты пакета PES в случае потока PES. Значение ES_rate действует в пакете PES, в котором оно включено в следующие пакеты PES того же самого потока PES до появления нового поля ES_rate. Значение скорости ES_rate измеряется в единицах по 50 байтов в секунду. Значение '0' является запрещенным. Значение скорости ES_rate используется для определения времени прибытия байтов на вход декодера P-STD для потоков PES, определенных в § 2.5.2.4. Значение, закодированное в поле ES_rat, может быть различным для различных пакетов PES_packet.

trick_mode_control – 3-битовое поле, которое указывает, какой из "хитрых" режимов применен для данного видеопотока. Для элементарных потоков других типов значение этого поля и все, что определяется следующими за ним пятью битами, является неопределенным. Статус поля trick_mode определен в разделе **"Хитрый" режим** параграфа 2.4.2.3.

Когда статус trick_mode = "false", количество раз N, когда кадр изображения оказывается на выходе процесса декодирования для последовательностей с прогрессивной разверткой, определяется для каждого кадра изображения полями repeat_first_field и top_field_first для потоков, соответствующих Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 "Видео", или указывается в заголовке последовательности для видеопотоков, соответствующих ИСО/МЭК 11172-2.

Для последовательностей с чересстрочной разверткой, когда статус trick_mode = "false", количество раз N, когда кадр изображения оказывается на выходе процесса декодирования для последовательностей с чересстрочной разверткой, определяется для каждого кадра изображения потоков, соответствующих Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 "Видео", полями repeat_first_field и progressive_frame.

Когда статус trick_mode = "true", количество раз воспроизведения кадра изображения зависит от величины N.

Когда значение этого поля изменяется или работа в "хитром" режиме заканчивается, могут появиться любые комбинации следующих событий:

- нарушение непрерывности временной базы;
- нарушение непрерывности декодирования;
- нарушение непрерывности счетчика непрерывности.

Таблица 2-24 – Управляющие значения "хитрого" режима

Значение	Описание
'000'	Ускоренное воспроизведение
'001'	Замедленное воспроизведение
'010'	Стоп-кадр
'011'	Ускоренное обратное воспроизведение
'100'	Замедленное обратное воспроизведение
'101'-'111'	Зарезервированы

В рамках "хитрого" режима, нештатная скорость декодирования и воспроизведения могут привести к тому, что значения некоторых полей, определенных в видеоданных элементарных потоков, окажутся некорректными. Точно также, могут оказаться недействительными семантические ограничения, наложенные на структуру секции. Данное исключение относится к следующим элементам синтаксиса видеоданных:

- bit_rate;
- vbv_delay;
- repeat_first_field;
- v_axis_positive;
- field_sequence;
- subcarrier;
- burst_amplitude;
- subcarrier_phase.

При работе в "хитром" режиме декодер не может использовать значения, закодированные в этих полях.

В нормативных документах не требуется, чтобы декодеры декодировали поле trick_mode_control. Однако, к декодерам, которые должны декодировать поле trick_mode_control, должны применяться следующие нормативные требования.

Ускоренное воспроизведение – поле trick_mode_control имеет значение '000'. В том случае, когда это значение присутствует, оно указывает видеопоток с укоренным воспроизведением и определяет значение следующих пяти битов в заголовке пакета PES. Бит intra_slice_refresh может быть установлен в '1', указывая возможность потери макроблоков, которые декодер может заменить макроблоками из предыдущих декодированных кадров изображения, расположенными в тех же местах кадра. Поле field_id, определенное в таблице 2-25, указывает, какое поле или какие поля должны быть воспроизведены, Поле frequency_truncation указывает, что в кадр может быть включен ограниченный набор коэффициентов. Возможные значения этого поля показаны в таблице 2-26.

Замедленное воспроизведение – поле trick_mode_control имеет значение ‘001'. В том случае, когда это значение присутствует, оно указывает видеопоток с замедленным воспроизведением и определяет значение следующих пяти битов в заголовке пакета PES. В случае последовательностей с прогрессивной разверткой, кадр изображения должен воспроизводиться в течение (N × rep_cntrl) кадров, где значение N определено выше.

В случае последовательностей с прогрессивной разверткой, соответствующих ИСО/МЭК 11172-2 "Видео" и Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 "Видео", кадр изображения должен воспроизводиться в течение (N × rep_cntrl) кадров.

В случае последовательностей с чересстрочной разверткой, соответствующих Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, кадр изображения воспроизводиться в течение времени поля (N × rep_cntrl). Если изображение имеет формат видеокадра с чересстрочной разверткой, то первым должно воспроизводиться верхнее поле, если top_field_first = 1, или нижнее поле, если top_field_first = '0' (см. Рекомендацию МСЭ-Т H.262 | ИСО/МЭК 13818-2). Это поле воспроизводится в течение (N × rep_cntrl / 2) полей. Затем воспроизводится другое поле этого кадра изображения в течение (N – N × rep_cntrl / 2) полей.

стоп-кадр – поле trick_mode_control имеет значение ‘010'. В том случае, когда это значение присутствует, оно указывает видеопоток, состоящий из стоп-кадров, и определяет значение следующих пяти битов в заголовке пакета PES. Поле field_id, определенное в таблице 2-25, указывает какое поле или какие поля должны быть воспроизведены. Поле field_id ссылается на первый модуль доступа видеопотока, который начинается в пакете PES, содержащем поле field_id, если только этот пакет PES не содержит ноль байтов полезной нагрузки. В последнем случае поле field_id ссылается на последний по времени предыдущий модуль доступа видеопотока.

Ускоренное обратное воспроизведение – поле trick_mode_control имеет значение ‘011'. В том случае, когда это значение присутствует, оно указывает видеопоток с ускоренным обратным воспроизведением и определяет значение следующих пяти битов в заголовке пакета PES. Бит intra_slice_refresh может быть установлен в '1', указывая возможность потери макроблоков, которые декодер может заменить макроблоками из предыдущих декодированных кадров изображения, расположенными в тех же местах кадра. Поле field_id, определенное в таблице 2-25, указывает, какое поле или какие поля должны быть воспроизведены, Поле frequency_truncation указывает, что в кадр может быть включен ограниченный набор коэффициентов. Значение величин этого поля показаны в таблице 2-26.

Замедленное обратное воспроизведение – поле trick_mode_control имеет значение ‘100'. В том случае, когда это значение присутствует, оно указывает видеопоток с замедленным обратным воспроизведением и определяет значение следующих пяти битов в заголовке пакета PES. В случае последовательностей с прогрессивной разверткой, соответствующих ИСО/МЭК 11172-2 "Видео" и Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 "Видео", кадр изображения должен воспроизводиться в течение (N × rep_cntrl) кадров, где значение N определено выше.

В случае последовательностей с чересстрочной разверткой, соответствующих Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, кадр изображения должен воспроизводиться в течение (N × rep_cntrl) полей. Если изображение имеет формат видеокадра с чересстрочной разверткой, то первым должно воспроизводиться нижнее поле, если значение top_field_first = 1, или верхнее поле, если значение top_field_first = '0' (см. Рекомендацию МСЭ-Т H.262 | ИСО/МЭК 13818-2). Это поле воспроизводится в течение (N × rep_cntrl / 2) полей. Затем воспроизводится другое поле этого кадра изображения в течение (N – N × rep_cntrl / 2) полей.

field_id – 2-битовое поле, которое указывает, какое поле или какие поля должны быть воспроизведены. Оно кодируется в соответствии с таблицей 2-25.

Таблица 2-25 – Управляющие значения поля field_id

Значение	Описание
'00'	Воспроизводить только верхнее поле
'01'	Воспроизводить только нижнее поле
'10'	Воспроизводить полный кадр
'11'	Зарезервированы

intra_slice_refresh – 1-битовый флаг, который, когда установлен в '1', указывает возможность потери макроблоков между кодированными секциями видеоданных в этом пакете PES. Когда он установлен в '0', потери макроблоков может не происходить. Более подробные сведения приведены в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2. Декодер может заменить потерянные макроблоки макроблоками из предыдущих декодированных кадров изображения, расположенными в тех же местах кадра.

frequency_truncation – 2-битовое поле, которое указывает, что при кодировании видеоданных в этом пакете PES может быть использован ограниченный набор коэффициентов. Значения определяются в таблице 2-26.

Таблица 2-26 – Выбор значений коэффициентов

Значение	Описание
'00'	Не нулевыми являются только коэффициенты DC
'01'	Не нулевыми являются только первые три коэффициента
'10'	Не нулевыми являются только первые шесть коэффициентов
'11'	Все коэффициенты могут быть не нулевыми

rep_cntrl – 5-битовое поле, которое указывает количество раз, которое должно быть воспроизведено каждое поле кадра изображения с чересстрочной разверткой, или количество раз, которое должен быть воспроизведен кадр изображения с прогрессивной разверткой. Какое поле – верхнее или нижнее – должно быть воспроизведено первым (для кадров изображения с чересстрочной разверткой) зависит от поля trick_mode_control и бита top_field_first в заголовке видеопоследовательности. Значение '0' является запрещенным.

additional_copy_info – Это 7-битовое поле содержит конфиденциальную информацию, относящуюся к информации защиты авторских прав.

previous_PES_packet_CRC – Поле previous_PES_packet_CRC – это 16-битовое поле, содержащее значение кода CRC, которое в результате приводит к получению нулей на выходе 16 регистров в декодере, аналогичном тому, который определен в Приложение A, но с полиномиальной характеристикой:

$$x^{16} + x^{12} + x^{5} + 1$$

после обработки байтов данных предыдущего пакета PES, за исключением заголовка пакета PES.

ПРИМЕЧАНИЕ 5. – Этот код CRC предназначен для использования при обслуживании сети, например, для изоляции источников регулярно повторяющихся ошибок. Он не предназначен для использования в декодерах элементарного потока. Он вычисляется только для байтов данных, поскольку данные заголовка пакета PES могут изменяться во время транспортировки.

PES_private_data_flag – 1-битовый флаг, который, когда установлен в '1', указывает, что заголовок пакета PES содержит конфиденциальную информацию. Когда он установлен в '0', он указывает, что конфиденциальная информация в заголовке PES отсутствует.

pack_header_field_flag – 1-битовый флаг, который, когда установлен в '1', указывает, что в заголовке этого блок-пакета PES содержится заголовок пакета, соответствующего ИСО/МЭК 11172-1, или заголовок пакета Программного потока. Если это поле присутствует в пакете PES, который содержится в Программном потоке, то оно должно быть установлено равным '0'. В Транспортном потоке, когда это поле установлено в '0', оно указывает, что в заголовке PES нет ни одного заголовка пакета.

program_packet_sequence_counter_flag – 1-битовый флаг, который, когда установлен в '1', указывает, что в этом пакете PES присутствуют поля program_packet_sequence_counter, MPEG1_MPEG2_identifier и original_stuff_length. Когда он установлен в '0', он указывает, что эти поля отсутствуют в заголовке PES.

P-STD_buffer_flag – 1-битовый флаг, который, когда установлен в '1' указывает, что в заголовке пакета PES присутствуют поля P-STD_buffer_scale и P-STD_buffer_size. Когда он установлен в '0', указывает, что эти поля в заголовке PES отсутствуют.

PES_extension_flag_2 – 1-битовое поле, которое, когда установлено в '1', указывает присутствие поля PES_extension_field_length и связанных с ним полей. Когда оно установлено в '0', это указывает, что поле PES_extension_field_length и любые связанные с ним поля отсутствуют.

PES_private_data – Это 16-байтовое поле, которое содержит конфиденциальную информацию. Эти данные, объединенные с данными, расположенными до и после этого поля, не должны эмулировать префикс стартового кода пакета packet_start_code_prefix (0x000001).

pack_field_length – Это 8-битовое поле, которое указывает длительность поля pack_header_field() в байтах.

program_packet_sequence_counter – Поле program_packet_sequence_counter это 7-битовое поле. Оно представляет собой дополнительный счетчик, который увеличивается на единицу с приходом каждого последовательного пакета PES программного потока или потока, соответствующего ИСО/МЭК 11172-1, или пакетов PES, связанных с определением отдельной программы в Транспортном потоке, обеспечивая выполнение действий, аналогичных функциям счетчика непрерывности (см. § 2.4.3.2). Это дает приложению возможность восстановить исходную последовательность пакетов PES Программного потока или исходную последовательность пакетов исходного потока, соответствующего ИСО/МЭК 11172-1. После достижения своего максимального значения, счетчик будет возвращен к 0. Пакеты PES не должны повторяться. Следовательно, никакие два последовательных пакета PES в мультиплексированном потоке программы не должны иметь одинаковых значений в поле program_packet_sequence_counter.

MPEG1_MPEG2_identifier – 1-битовый флаг, который, когда установлен в '1', указывает, что в этом пакете PES передается информация потока, соответствующего ИСО/МЭК 11172-1. Когда он установлен в '0', он указывает, что в этом пакете PES передается информация Программного потока.

original_stuff_length – Это 6-битовое поле определяет количество байтов заполнения, использованных в заголовке пакета PES исходного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, или заголовке пакета исходного потока, соответствующего ИСО/МЭК 11172-1.

P-STD_buffer_scale – Поле P-STD_buffer_scale – это 1-битовое поле, значение которого определено только, если этот пакет PES передается в Программном потоке. Оно указывает коэффициент масштабирования, использованный для описания следующего поля P-STD_buffer_size. Если предыдущий идентификатор потока stream_id указывает аудиопоток, то поле P-STD_buffer_scale должно иметь значение '0'. Если предыдущий идентификатор потока stream_id указывает видеопоток, то поле P-STD_buffer_scale должно иметь значение '1'. Для всех других типов потока это значение может быть либо '1', либо '0'.

P-STD_buffer_size – Поле P-STD_buffer_size – это 13-битовое целое число без знака, значение которого определено только, если этот пакет PES передается в Программном потоке. Оно определяет размер входного буфера BS_n в декодере P-STD. Если поле P-STD_buffer_scale имеет значение '0', то величина P-STD_buffer_size указывает размер буфера в единицах по 128 байтов. Если поле P-STD_buffer_scale имеет значение '1', то величина P-STD_buffer_size указывает размер буфера в единицах по 1024 байтов. Таким образом:

$$\begin{aligned} &\text{если } (P-STD_buffer_scale == 0) \\ &BS_n = \overline{P}-STD_buffer_size \times 128 \end{aligned} \tag{2-16}$$

в ином случае:

$$BS_n = P-STD_buffer_size \times 1024 \tag{2-17}$$

Закодированное значение размера буфера декодера P-STD вступает в силу сразу же после того, как поле P-STD_buffer_size принято декодером конечной системы, соответствующим Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 (см. § 2.7.7).

Размер буфера BS_n должен быть больше или равен размеру CPB, о котором сообщается в поле CpbSize[cpb_cnt_minus1], определенном значениями полей NAL hrd_parameters() в видеопотоке AVC. Если в видеопотоке AVC отсутствуют поля NAL hrd_parameters(), то размер буфера BS_n должен быть больше или равен размеру NAL CPB для формата потока байтов, определенного в Приложении A Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, как $1200 \times$ MaxCPB для используемого уровня.

PES_extension_field_length – Это 7-битовое поле, которое указывает длительность, в байтах, данных, следующих поля в поле расширения PES после этого и до появления каких-либо зарезервированных байтов, включительно.

stream_id_extension_flag – 1-битовый флаг, который, когда он установлен в значение '0', указывает, что в заголовке пакета PES присутствует поле stream_id_extension. Значение '1' для этого флага зарезервировано.

stream_id_extension – В программных потоках поле stream_id_extension задает тип и номер элементарного потока, как это определено значением поля stream_id_extension в таблице 2-27. В Транспортных потоках поле stream_id_extension может быть установлено равным любому допустимому значению, которое правильно описывает тип элементарного потока, согласно таблице 2-27. В Транспортных потоках тип элементарного потока задается в специальной информации программы, как указано в § 2.4.4. Следует отметить, что это поле используется как расширение определенного выше поля stream_id. Это поле не должно использоваться, если только значение идентификатора stream_id не равно 1111 1101.

Таблица 2-27 – Присвоение значений поля Stream_id_extension

stream_id_extension	Примечание	Кодировка потока
000 0000	1	Поток управляющей информации IPMP
000 0001	2	Поток IPMP
000 0010 … 011 1111		reserved_data_stream
100 0000 … 111 1111		private_stream
ПРИМЕЧАНИЕ 1. – Пакеты PES с полем stream_id_extension ‘0b000 0000’ (IPMP Control Information Stream – поток управляющей информации IPMP) имеют уникальный синтаксис, определенный в ИСО/МЭК 13818-11 (MPEG-2 IPMP). ПРИМЕЧАНИЕ 2. – Пакеты PES с полем stream_id_extension ‘0b000 0001’ (IPMP Stream – поток IPMP) имеют уникальный синтаксис, определенный в ИСО/МЭК 13818-11 (MPEG-2 IPMP).		

stuffing_byte – Это фиксированное 8-битовое значение, равное '1111 1111', которое может вводиться кодером, например, для удовлетворения требований канала. Оно отбрасывается декодером. Число байтов заполнения в заголовке одного пакета PES не должно превышать 32.

PES_packet_data_byte – Байты PES_packet_data_byte должны представлять собой непрерывный поток байтов данных элементарного потока, указанного идентификатором потока данного пакета (stream_id) или идентификатором PID. Когда данные элементарных потоков соответствуют Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 13818-3, байты PES_packet_data_byte должны быть побайтно синхронизированы с байтами, соответствующими настоящей Рекомендации | Международному стандарту. Порядок следования байтов элементарного потока должен быть сохранен. Количество байтов PES_packet_data_bytes (N) определяется в поле PES_packet_length. N должно быть равно значению, указанному в поле PES_packet_length, минус количество байтов между последним байтом поля PES_packet_length и первым байтом поля PES_packet_data_byte.

Для потоков private_stream_1, private_stream_2, ECM_stream или EMM_stream содержание поля PES_packet_data_byte определяется пользователем и в будущем в документах МСЭ-Т | ИСО/МЭК определяется не будет.

padding_byte – Это фиксированное 8-битовое значение, равное '1111 1111'. Оно отбрасывается декодером.

2.4.3.8 Передача Программных потоков и Системных потоков, соответствующих ИСО/МЭК 11172-1, в Транспортном потоке

Транспортный поток содержит дополнительные поля, предназначенные для обеспечения передачи Программных потоков и Системных потоков, соответствующих ИСО/МЭК 11172-1, таким образом, который позволяет выполнить простое восстановление соответствующего потока на декодере.

При помещении Программного потока внутрь Транспортного потока, пакеты PES Программного потока со значениями stream_id = private_stream_1, видеопакеты, соответствующие Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 11172-2, и аудиопакеты. соответствующие ИСО/МЭК 13818-3 или ИСО/МЭК 11172-3, передаются в пакетах Транспортного потока.

При восстановлении Программного потока в декодере Транспортного потока данные этих пакетов PES копируются в восстанавливаемый Программный поток.

Для пакетов PES Программного потока, в которых значения stream_id равны program_stream_map, padding_stream, private_stream_2, ECM, EMM, DSM_CC_stream или program_stream_directory, все байты пакета PES Программного потока, за исключением префикса packet_start_code_prefix, помещаются в поля data_bytes нового пакета PES. Идентификатор потока stream_id этого нового пакета PES имеет значение ancillary_stream (см. таблицу 2-22). Затем этот новый пакет PES передается в пакетах Транспортного потока.

При восстановлении Программного потока в декодере Транспортного потока, для пакетов PES, у которых stream_id = ancillary_stream_id, префикс packet_start_code_prefix записывается в восстанавливаемый Программный поток, после него следуют поля data_byte из пакетов PES этого Транспортного потока.

Потоки, соответствующие ИСО/МЭК 11172-1, передаются внутри Транспортных потоков путем замены заголовков пакетов, соответствующих ИСО/МЭК 11172-1, заголовками пакетов PES, соответствующими Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2. Значения полей заголовков пакетов, соответствующих ИСО/МЭК 11172-1, копируются в эквивалентные поля заголовков пакетов PES, соответствующих Рекомендации. МСЭ-Т H.262 | ИСО/МЭК 13818-2.

Поле program_packet_sequence_counter вводится в заголовок каждого пакета PES, в котором передаются данные Программного потока или Системного потока, соответствующего ИСО/МЭК 11172-1. Это позволяет восстановить на декодере порядок следования пакетов PES в исходном Программном потоке, или в исходном Системном потоке, соответствующем ИСО/МЭК 11172-1.

Поле заголовка блок-пакета pack_header() Программного потока или Системного потока, соответствующего ИСО/МЭК 11172-1, передается в Транспортном потоке в заголовке, который следует сразу же после пакета PES.

2.4.4 Специальная информация программы

Специальная информация программы (PSI) содержит и нормативную информацию, соответствующую Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, и конфиденциальную информацию, которая позволяет выполнить в декодерах демультиплексирование программ. Программы состоят из одного или нескольких элементарных потоков, каждый из которых идентифицирован своим PID. Для реализации условного доступа программы, элементарные потоки или их части могут быть скремблированы. Однако, Специальная информация программы не должна подвергаться скремблированию.

В Транспортных потоках Специальная информация программы делится на шесть табличных структур, как показано в таблице 2-28. Хотя эти структуры могут считаться простыми таблицами, они должны быть разделены на секции и введены в пакеты Транспортного потока, некоторые с заранее определенными значениями PID, а другие со значениями PID, которые выбирает пользователь.

Таблица 2-28 – Специальная информация программы

Название структуры	Тип потока	Зарезервированный номер PID	Описание
Таблица ассоциаций для программы	Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1	0x00	Связывает номер программы и PID Таблицы преобразования программы
Таблица преобразования программы	Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1	Присвоен в PAT	Задает значения PID для компонентов одной или нескольких программ
Таблица сетевой информации	Конфиденциальный	Присвоен в PAT	Физические параметры сети, такие как частоты FDM, номера транспондеров и т. д.
Таблица условного доступа	Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1	0x01	Присваивает одному или каждому из нескольких (частных) потоков EMM уникальное значение PID
Таблица описания транспортного потока	Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1	0x02	Связывает один или несколько дескрипторов из таблицы 2-39 со всем транспортным потоком
Таблица управляющей информации IPMP	Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1	0x03	Содержит Список инструментов IPMP, Контейнер прав, Контейнер инструментов, определенные в стандарте ИСО/МЭК 13818-11

Таблицы PSI, определенные в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, должны быть разделены на одну или несколько секций, которые передаются в пакетах Транспортного потока. Секция – это синтаксическая структура, которая должна использоваться для преобразования каждой таблицы, определенной в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, в пакеты Транспортного потока.

Вместе с таблицами PSI, определенными в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, можно передавать таблицы конфиденциальной информации. Средства, при помощи которых конфиденциальная информация передается внутри пакетов Транспортного потока, в настоящей Спецификации не определяются. Она может быть структурирована тем же способом, который используется для передачи таблиц PSI, определенных в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, так, что синтаксис для преобразования этой конфиденциальной информации, идентичен синтаксису, используемому для преобразования информации

таблиц PSI, определенных в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Для этих целей определяется конфиденциальная секция (private_section). Если конфиденциальная информация передается в пакетах Транспортного потока с тем же значением PID, что и у пакетов Транспортного потока, в которых передаются таблицы преобразования программы (как определено в Таблице ассоциаций для программы), то должны использоваться синтаксис и семантика конфиденциальной секции (private_section). Данные, передаваемые в байтах private_data_bytes, могут быть скремблированными. Однако, никакие другие поля секции private_section скремблироваться не должны. Эта секция private_section позволяет передавать данные с использованием минимальной структуры. Когда эта структура не используется, настоящей Рекомендацией | Международным стандартом не определяется преобразование конфиденциальной информации в пакетах Транспортного потока.

Длина секций может быть переменной. Начало секции обозначается полем pointer_field в полезной нагрузке пакета Транспортного потока. Синтаксис этого определен в таблице 2-29.

В пакетах Транспортного потока, в которых передаются секции PSI, могут появляться поля адаптации.

В Транспортном потоке, в полезной нагрузке пакетов Транспортного потока, в которых передаются таблицы PSI и/или секции private_section, байты заполнения пакета, имеющие значение 0xFF могут появляться только после последнего байта такой секции. В таком случае все байты до окончания пакета Транспортного потока также должны быть байтами заполнения, имеющими значение 0xFF. Эти байты могут быть отброшены декодером. В таком случае полезная нагрузка следующего пакета Транспортного потока с тем же значением PID должна начинаться с поля pointer_field, имеющего значение 0x00, которое указывает, что сразу после него начинается следующая секция.

Каждый Транспортный поток должен содержать один или несколько пакетов Транспортного потока со значением PID = 0x0000. Все вместе эти пакеты Транспортного потока должны содержать полную Таблицу ассоциаций для программы, содержащую полный список всех программ в рамках Транспортного потока. К текущим данным, передаваемым в Транспортном потоке, должна всегда применяться самая последняя по времени переданная версия этой таблицы с индикатором current_next_indicator, установленным в значение '1'. Любые изменения в программах, передаваемых внутри Транспортного потока, должны быть описаны в обновленной версии Таблицы ассоциаций для программы (PAT), передаваемой в пакетах Транспортного потока, имеющих PID = 0x0000. В этих секциях должно использоваться значение table_id = 0x00. В пакетах Транспортного потока с PID = 0x0000 могут передаваться только секции с таким значением table_id. Для того чтобы новая версия таблицы PAT начала использоваться, все секции (указанные в поле last_section_number) с новым значением version_number и с текущим индикатором current_next_indicator, установленным в '1', должны покинуть буфер B_{sys}, определенный в декодере T-STD (см. § 2.4.2). Таблица PAT начинает использоваться, когда буфер B_{sys} покидает последний байт секции, необходимой для получения полной таблицы.

Если в Транспортном потоке один или несколько элементарных потоков скремблированы, то пакеты Транспортного потока, имеющие значение PID = 0x0001, должны передаваться с содержанием в них полной Таблицы условного доступа, включая дескрипторы CA_descriptor, связанные со скремблированными потоками. Передаваемые пакеты Транспортного потока все вместе образуют одну полную версию таблицы условного доступа. К текущим данным Транспортного потока должна всегда применяться версия этой таблицы, переданная последней по времени, с индикатором current_next_indicator, установленным в значение '1'. Любые изменения скремблирования, которые делают существующую таблицу недостоверной или неполной, должны быть описаны в обновленной версии таблицы условного доступа. Во всех этих секциях должно должен использоваться идентификатор table_id = 0x01. В пакетах Транспортного потока с PID = 0x0001 могут передаваться только секции с таким значением table_id. Для того чтобы начала использоваться новая версия таблицы CAT, все секции (указанные в поле last_section_number) с новым значением version_number и с текущим индикатором current_next_indicator, установленным в '1', должны покинуть буфер B_{sys}. Таблица CAT начинает использоваться, когда буфер B_{sys} покидает последний байт секции, необходимой для получения полной таблицы.

Каждый Транспортный поток должен содержать один или несколько пакетов Транспортного потока со значениями PID, которые в Таблице ассоциаций для программы обозначены как Пакеты Транспортного потока, содержащие секции преобразования программы TS. Каждая программа, перечисленная в Таблице ассоциаций для программы, должна быть описана в уникальной секции TS_program_map_section. Каждая программа должна быть полностью определена внутри самого Транспортного потока. Конфиденциальная информация, с которой связано соответствующее поле elementary_PID в соответствующей секции Таблицы преобразования программы, является частью программы. В Транспортном потоке может присутствовать другая конфиденциальная информация, которая не указана в секции Таблицы преобразования программы. К текущим данным Транспортного потока должна всегда применяться та версия секции TS_program_map_section, которая была передана последней по времени, с индикатором current_next_indicator, установленным в значение '1'. Любые изменения в определении любой из программ, передаваемых в Транспортном потоке, должны быть описаны в обновленной версии соответствующей секции Таблицы преобразования программы, передаваемой в пакетах Транспортного потока со значением PID, определенным как program_map_PID для этой конкретной программы. Все Пакеты Транспортного потока, в которых передается данная секция TS_program_map_section,

должны иметь одно и то же значение PID. На протяжении всего времени непрерывного наличия программы, включая все связанные с нею события, значение program_map_PID не должно изменяться. Определение программы не должно распространяться более чем на одну секцию TS_program_map_section. Новая версия секции TS_program_map_section начинает использоваться, когда буфер B_{sys} покидает последний байт этой секции с новым полем version_number текущим индикатором current_next_indicator, установленным в '1'.

Секции, имеющие значение table_id = 0x02, должны содержать информацию Таблицы преобразования программы. Такие секции могут передаваться в пакетах Транспортного потока с различными значениями PID.

Таблица сетевой информации является дополнительной, и ее содержание является конфиденциальным. Если она присутствует, то она передается в пакетах Транспортного потока, которые имеют одинаковое значение PID, которое называется network_PID. Значение network_PID определяется пользователем и, когда присутствует, должно находиться в Таблице ассоциаций для программы в зарезервированном значении program_number 0x0000. если существует таблица сетевой информации, она должна принимать вид одной или нескольких секций private_section.

Максимальное количество байтов в секции таблицы PSI, определенной в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, составляет 1024 байтов. Максимальное количество байтов в секции private_section составляет 4096 байтов.

Таблица описания транспортного потока является дополнительной. Если она присутствует, то описание Транспортного потока передается в пакетах Транспортного потока, которые имеют значение PID = 0x0002, определенное в Таблице 2-28, и должно применяться ко всему Транспортному потоку целиком. В разделах описания Транспортного потока должен использоваться идентификатор table_id = 0x03, как определено в таблице 2-31, и содержание этих разделов ограничивается дескрипторами, определенными в таблице 2-45. Секция TS_description_section начинает использоваться, когда буфер B_{sys} покидает последний байт секции, необходимой для получения полной таблицы.

Нет никаких ограничений на время появления стартовых кодов, байтов синхронизации и других комбинаций битов в информации PSI, которая определена в настоящей Рекомендации | Международном стандарте, или является конфиденциальной.

2.4.4.1 Указатель

Синтаксис поля pointer_field определяется в таблице 2-29.

Таблица 2-29 – Указатель Специальной информации программы

Синтаксис	Количество битов	Мнемоника
pointer_field	**8**	**uimsbf**

2.4.4.2 Семантическое определение полей в синтаксисе указателя

pointer_field – Это 8-битовое поле, значением которого должно быть количество байтов, которые следуют непосредственно после поля pointer_field до первого байта первой секции, которая присутствует в полезной нагрузке пакета Транспортного потока (таким образом, значение 0x00 поля pointer_field указывает, что секция начинается сразу же после поля pointer_field). Когда, как минимум, одна секция начинается в данном пакете Транспортного потока, то индикатор payload_unit_start_indicator (см. § 2.4.3.2) должен быть установлен равным '1', и первый байт полезной нагрузки этого пакета Транспортного потока должен содержать указатель. Когда ни одна секция не начинается в данном пакете Транспортного потока, то индикатор payload_unit_start_indicator должен быть установлен равным '0', и ни один указатель не должен передаваться в полезной нагрузке этого пакета.

2.4.4.3 Таблица ассоциаций для программы

Таблица ассоциаций для программы определяет соответствие между меткой program_number и значением PID для пакетов Транспортного потока, в которых передается определение программы. Метка program_number – это числовая метка, связанная с программой.

Общая таблица содержится в одной или нескольких секциях с описанным далее синтаксисом. Она может быть сегментирована и может занимать несколько секций (см. таблицу 2-30).

Таблица 2-30 – Секция ассоциаций для программы

Синтаксис	Количество битов	Мнемоника
program_association_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
transport_stream_id	**16**	**uimsbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
program_number	**16**	**uimsbf**
reserved	**3**	**bslbf**
if (program_number = = '0') {		
network_PID	**13**	**uimsbf**
}		
else {		
program_map_PID	**13**	**uimsbf**
}		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.4 Присваиваемые значения Table_id

Поле table_id определяет содержание секции PSI Транспортного потока, как показано в таблице 2-31.

Таблица 2-31 – Присваиваемые значения table_id

Значение	Описание
0x00	program_association_section
0x01	conditional_access_section (CA_section)
0x02	TS_program_map_section
0x03	TS_Description_section
0x04	ИСО_IEC_14496_scene_Description_section
0x05	ИСО_IEC_14496_object_descriptor_section
0x06	Metadata_section
0x07	IPMP_Control_Informatio_section (определенный в ИСО/МЭК 13818-11)
0x08-0x3F	Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 зарезервировано
0x40-0xFE	Частный идентификатор пользователя
0xFF	Запрещен

2.4.4.5 Семантические определения полей в секции ассоциаций для программы

table_id – Это 8-битовое поле, которое должно быть установлено равным 0x00, как показано в таблице 2-31.

section_syntax_indicator – Поле section_syntax_indicator – это 1-битовое поле, которое должно быть установлено равным '1'.

section_length – Это 12-битовое поле, первые два бита которого должны иметь значение '00'. Оставшиеся 10 битов определяют количество байтов в секции, начинающейся сразу же после поля section_length, включая CRC. Значение в этом поле не должно превышать величины 1021 (0x3FD).

transport_stream_id – Это 16-битовое поле, которое служит меткой, для того чтобы отличить этот Транспортный поток от любого другого мультиплексированного потока в сети. Его значение определяется пользователем.

version_number – Это 5-битовое поле представляет собой номер версии полной Таблицы ассоциаций для программы. Номер версии должен увеличиваться на 1 (по модулю 32) всегда, когда изменяется Таблица ассоциаций для программы. Когда индикатор current_next_indicator установлен в '1', поле version_number должно соответствовать применяемой в текущий момент версии Таблицы ассоциаций для программы. Когда индикатор current_next_indicator установлен в '0', поле version_number должно соответствовать следующей применяемой версии Таблицы ассоциаций для программы.

current_next_indicator – 1-битовый индикатор, который, когда установлен в '1', указывает, что в настоящий момент применяется переданная Таблица ассоциаций для программы. Когда этот бит установлен в '0', он указывает, что переданная таблица уже неприменима, и что должна быть введена в действие следующая таблица.

section_number – Это 8-битовое поле указывает номер этой секции. Поле section_number первой секции в Таблице ассоциаций для программы должно иметь значение 0x00. Оно должно увеличиваться на 1 с появлением каждой дополнительной секции в Таблице ассоциаций для программы.

last_section_number – Это 8-битовое поле указывает номер последней секции (то есть, секции с наибольшим номером section_number) полной Таблицы ассоциаций для программы.

program_number – Поле program_number – это 16-битовое поле. Он определяет программу, к которой применяется поле program_map_PID. Когда он установлен в 0x0000, то следующим опорным PID должен быть PID сети. Во всех других случаях значение этого поля определяется пользователем. Это поле не должно принимать никакого значения более чем один раз в пределах одной версии Таблицы ассоциаций для программы.

ПРИМЕЧАНИЕ. – Поле program_number может использоваться, например, для обозначения радиовещательного канала.

network_PID – Поле network_PID – это 13-битовое поле, которое используется только в объединении со значением поля program_number = 0x0000 и определяет значение PID пакетов Транспортного потока, которые должны содержать Таблицу сетевой информации. Значение поля network_PID определяется пользователем, но оно должно принимать только значения, определенные в таблице 2-3. Присутствие идентификатора network_PID является необязательным.

program_map_PID – Поле program_map_PID – это 13-битовое поле, определяющее значение PID пакетов Транспортного потока, которые должны содержать секцию program_map_section, применимую программе, определенной полем program_number. Ни одно поле program_number не должно иметь более одного назначения program_map_PID. Значение поля program_map_PID определяется пользователем, но оно должно принимать только значения, определенные в таблице 2-3.

CRC_32 – Это 32-битовое поле, которое содержит значение кода CRC, которое в результате приводит к получению нулей на выходе регистров в декодере, определенного в Приложении A, после обработки полной секции ассоциаций для программы.

2.4.4.6 Таблица условного доступа

Таблица условного доступа (CA) описывает связь между одной или несколькими системами условного доступа, их потоки EMM и любые конкретные параметры, связанные с ними. Описание поля descriptor() в таблице 2-32 приводится в § 2.6.16.

Таблица состоит их одной или нескольких секций с нижеприведенным синтаксисом. Она может быть сегментирована и занимать несколько секций.

Таблица 2-32 – Секция условного доступа

Синтаксис	Количество битов	Мнемоника
CA_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
зарезервировано	**2**	**bslbf**
section_length	**12**	**uimsbf**
зарезервировано	**18**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.7 Семантические определения полей в секции условного доступа

table_id – Это 8-битовое поле, которое должно быть установлено равным 0x01, как определено в таблице 2-31.

section_syntax_indicator – Индикатор section_syntax_indicator – это 1-битовое поле, которое должно быть установлено равным '1'.

section_length – Это 12-битовое поле, первые два бита которого должны иметь значение ‘00'. Оставшиеся 10 битов определяют количество байтов в секции, которая начинается сразу же после поля section_length, включая CRC. Значение в этом поле не должно превышать величины 1021 (0x3FD).

version_number – Это 5-битовое поле представляет собой номер версии полной таблицы условного доступа. Номер версии должен увеличиваться на 1 (по модулю 32), каждый раз, когда происходит изменение информации, передаваемой в таблице CA. Когда индикатор current_next_indicator установлен в '1', поле version_number должно соответствовать применяемой в текущий момент версии Таблицы условного доступа. Когда индикатор current_next_indicator установлен в '0', поле version_number должно соответствовать следующей применяемой версии Таблицы условного доступа.

current_next_indicator – 1-битовый индикатор, который, когда установлен в '1' указывает, что в настоящий момент применяется переданная Таблица условного доступа. Когда этот бит установлен в '0', он указывает, что переданная Таблица условного доступа уже неприменима, и что должна быть введена в действие следующая Таблица условного доступа.

section_number – Это 8-битовое поле указывает номер этой секции. Поле section_number первой секции в Таблице условного доступа должно иметь значение 0x00. Оно должно увеличиваться на 1 с введением каждой дополнительной секции в Таблицу условного доступа.

last_section_number – Это 8-битовое поле указывает номер последней секции (то есть, секции с наибольшим номером section_number) полной Таблицы условного доступа.

CRC_32 – Это 32-битовое поле, которое содержит значение кода CRC, которое в результате приводит к получению нулей на выходе регистров в декодере, определенного в Приложении A, после обработки полной секции условного доступа.

2.4.4.8 Таблица преобразования программы

Таблица преобразования программы обеспечивает преобразование между номерами программ и элементами программы, которые их составляют. Отдельный экземпляр такого преобразования называется "определением программы". Таблица преобразования программы представляет собой полное собрание всех определений программы для Транспортного потока. Эта таблица должна передаваться в пакетах, значения PID для которых выбираются кодером. При желании может использоваться несколько значений PID. Эта таблица содержится в одной или нескольких секциях с описанным ниже синтаксисом. Она может быть сегментирована и занимать несколько секций. В каждой секции номер поля должен быть установлен, равным нулю. Секции определяются полем номера программы program_number.

Определение полей дескрипторов descriptor() приводятся в § 2.6 (см. таблицу 2-33).

Таблица 2-33 – Секция преобразования программы для Транспортного потока

Синтаксис	Количество битов	Мнемоника
TS_program_map_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
зарезервировано	**2**	**bslbf**
section_length	**12**	**uimsbf**
program_number	**16**	**uimsbf**
зарезервировано	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
зарезервировано	**3**	**bslbf**
PCR_PID	**13**	**uimsbf**
зарезервировано	**4**	**bslbf**
program_info_length	**12**	**uimsbf**
for (i = 0; i < N; i++) {		

Синтаксис	Количество битов	Мнемоника
descriptor()		
}		
for (i = 0; i < N1; i++) {		
stream_type	**8**	**uimsbf**
зарезервировано	**3**	**bslbf**
elementary_PID	**13**	**uimsbf**
зарезервировано	**4**	**bslbf**
ES_info_length	**12**	**uimsbf**
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.9 Семантические определения полей в секции преобразования программы в Транспортном потоке

table_id – Это 8-битовое поле, которое для секции TS_program_map_section должно быть всегда установлено в значение 0x02, как показано в Таблице 2-31.

section_syntax_indicator – Индикатор section_syntax_indicator – это 1-битовое поле, которое должно быть установлено равным '1'.

section_length – Это 12-битовое поле, первые два бита которого должны иметь значение ‘00'. Оставшиеся 10 битов определяют количество байтов в секции, которая начинается сразу же после поля section_length, включая CRC. Значение в этом поле не должно превышать величины 1021 (0x3FD).

program_number – Поле program_number – это 16-битовое поле. Оно определяет программу, к которой применим идентификатор program_map_PID. Одно определение программы должно передаваться только в одной секции TS_program_map_section. Это означает, что определение программы никогда не бывает длиннее, чем 1016 (0x3F8). Способы работы в случаях, когда этой длины недостаточно, описаны в Информативном Приложении C. Поле program_number может использоваться, например, в качестве обозначения радиовещательного канала. При помощи описания различных элементов программы, которые принадлежат этой программе, можно объединить вместе данные из различных источников (например, последовательные события), формируя с помощью поля program_number непрерывный набор потоков. Примеры применения описаны в Приложении C.

version_number – Это 5-битовое поле представляет собой номер версии секции TS_program_map_section. Номер версии должен увеличиваться на 1 (по модулю 32), каждый раз, когда происходит изменение информации, передаваемой в этой секции. Номер версии ссылается на определение отдельной программы, и, следовательно, отдельной секции. Когда индикатор current_next_indicator установлен в '1', поле version_number должно соответствовать применяемой в текущий момент версии секции TS_program_map_section. Когда индикатор current_next_indicator установлен в '0', поле version_number должно соответствовать следующей применяемой версии секции TS_program_map_section.

current_next_indicator –1-битовое поле, которое, когда установлено в '1' указывает, что в настоящий момент применяется переданная секция TS_program_map_section. Когда этот бит установлен в ‘0’, он указывает, что переданная секция TS_program_map_section уже неприменима, и что должна быть введена в действие следующая секция TS_program_map_section.

section_number – Значение этого 8-битового поля должно быть равно 0x00.

last_section_number – Значение этого 8-битового поля должно быть равно 0x00.

PCR_PID – Это 13-битовое поле, указывающее значение PID пакетов Транспортного потока, которые должны содержать поля PCR, действующие для программы, определенной полем program_number. Если для потоков конфиденциальной информации ни одно значение PCR не связано с определением программы, то поле PCR_PID должно принимать значение 0x1FFF. Семантические определения PCR приводятся в § 2.4.3.5, а ограничения, накладываемые на выбор значения PCR_PID, показаны в таблице 2-3.

program_info_length – Это 12-битовое поле, первые два бита которого должны иметь значение '00'. Оставшиеся 10 битов определяют количество байтов в дескрипторах, которые следуют сразу же после поля program_info_length.

stream_type – Это 8-битовое поле, определяющее тип элемента программы, передаваемого в пакетах с PID, значение, которое определяется полем elementary_PID. Значения поля stream_type определяются в таблице 2-34.

ПРИМЕЧАНИЕ. – Вспомогательный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, также может использоваться в типах данных, определяемых в настоящей Спецификации, отличных от аудио, видео и DSM-CC, например, в Директории Программного потока или Карте преобразований Программного потока.

Таблица 2-34 – Присвоение типов потоков

Значение	Описание
0x00	Зарезервирован в МСЭ-Т | ИСО/МЭК
0x01	Видео в соответствии с ИСО/МЭК 11172
0x02	Видео в соответствии с Рек. МСЭ-Т H.262 | ИСО/МЭК 13818-2 или видеопоток с ограниченными параметрами в соответствии с ИСО/МЭК 11172-2
0x03	Аудио в соответствии с ИСО/МЭК 11172-3
0x04	Аудио в соответствии с ИСО/ МЭК 13818-3
0x05	Конфиденциальные разделы (private_sections) в соответствии с Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1
0x06	Пакеты PES, содержащие конфиденциальную информацию в соответствии с Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1
0x07	MHEG в соответствии с ИСО/МЭК 13522
0x08	DSM-CC в соответствии с Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 Приложение A
0x09	Рек. МСЭ-Т H.222.1
0x0A	ИСО/МЭК 13818-6 тип A
0x0B	ИСО/МЭК 13818-6 тип B
0x0C	ИСО/МЭК 13818-6 тип C
0x0D	ИСО/МЭК 13818-6 тип D
0x0E	Дополнительные данные в соответствии с Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1
0x0F	Аудио с синтаксисом транспорта ADTS в соответствии с ИСО/МЭК 13818-7
0x10	Визуальная информация в соответствии с ИСО/МЭК 14496-2
0x11	Аудио в соответствии с ИСО/МЭК 14496-3 с синтаксисом транспорта LATM, определенным в ИСО/МЭК 14496-3
0x12	Пакетированный поток SL в соответствии с ИСО/МЭК 14496-1 или поток FlexMux, передаваемый в пакетах PES
0x13	Пакетированный поток SL в соответствии с ИСО/МЭК 14496-1 или поток FlexMux, передаваемый в секциях ИСО/МЭК 14496_sections
0x14	ИСО/МЭК 13818-6 Протокол синхронизированной загрузки
0x15	Метаданные, передаваемые в пакетах PES
0x16	Метаданные, передаваемые в секциях metadata_sections
0x17	Метаданные, передаваемые в данных карусельного типа в соответствии с ИСО/МЭК 13818-6
0x18	Метаданные, передаваемые в объекте карусельного типа в соответствии с ИСО/МЭК 13818-6
0x19	Метаданные, передаваемые в Протоколе синхронизированной загрузки в соответствии с ИСО/МЭК 13818-6
0x1A	Поток IPMP (определен в ИСО/МЭК 13818-11, MPEG-2 IPMP)
0x1B	Видеопоток AVC, определенный в Рек. МСЭ-Т H.264 | ИСО/МЭК 14496-10
0x1C-0x7E	Зарезервирован в соответствии с Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1
0x7F	Поток 0IPMP m
0x80-0xFF	Конфиденциальный поток пользователя

elementary_PID – Это 13-битовое поле, определяющее значение PID пакетов Транспортного потока, в которых передается элемент соответствующей программы.

ES_info_length – Это 12-битовое поле, первые два бита которого должны иметь значение '00'. Оставшиеся 10 битов определяют количество байтов описателей элемента соответствующей программы, который следует сразу же после поля ES_info_length.

CRC_32 – Это 32-битовое поле, которое содержит значение кода CRC, которое в результате приводит к получению нулей на выходе регистров в декодере, определенном в Приложении B, после обработки байтов данных целой секции карты программ Транспортного потока.

2.4.4.10 Синтаксис секции конфиденциальной информации

Когда конфиденциальная информация передается в пакетах Транспортного потока, имеющих значение PID, обозначенное в Таблице ассоциаций для программы как PID Таблицы преобразования программы, должна

использоваться секция private_section. Секция private_section позволяет передавать данные с минимальной структурой, и, тем не менее, позволяет декодеру провести синтаксический анализ потока. Эти секции могут использоваться двумя способами: если индикатор section_syntax_indicator установлен в '1', то должна использоваться вся структура, общая для всех таблиц; если этот индикатор установлен в '0', то общему синтаксису и семантике структуры должны соответствовать только поля 'table_id' – 'private_section_length', а остальная часть секции private_section может принимать любой вид, определенный пользователем. Примеры расширенного использования этого синтаксиса приведены в информативном Приложении C.

Конфиденциальная таблица может быть сформирована из нескольких секций private_sections, причем все они имеют один и тот же идентификатор table_id (см. таблицу 2-35).

Таблица 2-35 – Секция конфиденциальной информации

Синтаксис	Количество битов	Мнемоника
private_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
Зарезервированы	**2**	**bslbf**
private_section_length	**12**	**uimsbf**
if (section_syntax_indicator = = '0') {		
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**bslbf**
}		
}		
else {		
table_id_extension	**16**	**uimsbf**
Зарезервированы	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < private_section_length-9; i++) {		
private_data_byte	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		
}		

2.4.4.11 Семантические определения полей в секции конфиденциальной информации

table_id – Это 8-битовое поле, значение в котором определяет Таблицу конфиденциальной информации, которой принадлежит эта секция. Могут использоваться только значения, определенные в строке "Частный идентификатор пользователя " Таблицы 2-31

section_syntax_indicator – Это 1-битовый индикатор. Когда он установлен в '1', он указывает, что вне рамок, определенных полем private_section_length, секция конфиденциальной информации соответствует общему синтаксису секции. Когда он установлен в '0', он указывает, что байты private_data_byte следуют сразу же после поля private_section_length.

private_indicator – Это 11-битовый флаг, определяемый пользователем, который в будущем не должен определяться документами МСЭ-Т | ИСО/МЭК.

private_section_length – 12-битовое поле. Оно определяет количество оставшихся байтов в секции конфиденциальной информации, которые следуют сразу же после поля private_section_length до окончания секции private_section. Значение в этом поле не должно превышать величины 4093 (0xFFD).

private_data_byte – Поле private_data_byte определяется пользователем и в будущем не должно определяться документами МСЭ-Т | ИСО/МЭК.

table_id_extension – Это 16-битовое поле. Его использование и его значение определяются пользователем.

version_number – Это 5-битовое поле представляет собой номер версии секции private_section. Номер версии должен увеличиваться на 1 (по модулю 32) каждый раз, когда происходит изменение информации, передаваемой в секции private_section. Когда индикатор current_next_indicator установлен в '0', поле version_number должно соответствовать следующей применяемой версии секции private_section с теми же самыми значениями table_id и section_number.

current_next_indicator – Это 1-битовое поле, которое, когда установлено в '1' указывает, что в настоящий момент применяется переданная секция private_section. Когда индикатор current_next_indicator установлен в '1', поле version_number должно соответствовать применяемой в текущий момент версии секции private_section. Когда этот бит установлен в ‘0’, он указывает, что переданная секция private_section уже неприменима, и что должна быть введена в действие следующая секция private_section с теми же самыми значениями section_number и table_id.

section_number – Это 8-битовое поле указывает номер секции private_section. Поле section_number первой секции в Таблице конфиденциальной информации должно иметь значение 0x00. С введением каждой дополнительной секции в Таблицу конфиденциальной информации поле section_number должно увеличиваться на 1.

last_section_number – Это 8-битовое поле указывает номер последней секции (то есть, секции с наибольшим номером section_number) Таблицы конфиденциальной информации, к которой принадлежит данная секция.

CRC_32 – Это 32-битовое поле, содержащее значение кода CRC, которое в результате приводит к получению нулей на выходе регистров в декодере, определенного в Приложении A, после обработки полной секции конфиденциальной информации.

2.4.4.12 Синтаксис секции Транспортного потока

Двоичные потоки, соответствующие Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, могут передавать информацию, определенную в Таблице 2-36. Декодеры, соответствующие Рекомендации Рек. МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, могут декодировать информацию, определенную в этой таблице.

Таблица описания Транспортного потока определяется для обеспечения передачи дескрипторов, как описано в § 2.6 для всего Транспортного потока в целом. Дескрипторы должны применяться ко всему Транспортному потоку в целом. В этой таблице используется значение table_id = 0x03, определенное в таблице 2-31, и оно передается в пакетах Транспортного потока, имеющих значение PID = 0x0002, как определено в таблице 2-3.

Таблица 2-36 – Таблица описания транспортного потока

Синтаксис	Количество битов	Мнемоника
TS_Description_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
'0'	**1**	**bslbf**
Зарезервированы	**2**	**bslbf**
section_length	**12**	**uimsbf**
Зарезервированы	**18**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	**32**	**rpchof**
}		

2.4.4.13 Семантические определения полей в секции Транспортного потока

table_id – Это 8-битовое поле, которое должно быть установлено равным '0x03', как определено в таблице 2-31.

section_length – Это 12-битовое поле, первые два бита которого должны иметь значение ‘00'. Оставшиеся 10 битов определяют количество байтов в секции, которая начинается сразу же после поля section_length, включая CRC. Значение в этом поле не должно превышать величины 1021 (0x3FD).

version_number – Это 5-битовое поле представляет собой номер версии всей Таблицы описания транспортного потока в целом. Номер версии должен увеличиваться на 1 (по модулю 32) всегда, когда меняется определение

Таблицы описания транспортного потока. Когда индикатор current_next_indicator установлен в '1', то поле version_number должно соответствовать применяемой в текущий момент Таблице описания транспортного потока. Когда индикатор current_next_indicator установлен в '0', поле version_number должно соответствовать следующей применяемой версии Таблицы описания транспортного потока.

current_next_indicator – 1-битовый индикатор, который, когда установлен в '1', указывает, что в настоящий момент применяется переданная Таблица описания транспортного потока. Когда этот бит установлен в ‘0’, он указывает, что переданная таблица уже неприменима, и что должна быть введена в действие следующая таблица.

section_number – Это 8-битовое поле указывает номер секции. Поле section_number первой секции в Таблице описания транспортного потока должно иметь значение 0x00. Оно должно увеличиваться на 1 с введением каждой дополнительной секции в Таблицу описания транспортного потока.

last_section_number – Это 8-битовое поле указывает номер последней секции (то есть, секции с наибольшим номером section_number) полной Таблицы описания транспортного потока.

CRC_32 – Это 32-битовое поле, которое содержит значение кода CRC, которое в результате приводит к получению нулей на выходе регистров в декодере, определенного в Приложении A, после обработки полной секции описания транспортного потока.

2.5 Требования к двоичному программному потоку

2.5.1 Структура и параметры кодирования Программного потока

Уровень кодирования Программного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, позволяет объединять в отдельный поток одну программу из одного или нескольких элементарных потоков. Данные из каждого элементарного потока мультиплексируются вместе с информацией, которая обеспечивает синхронное воспроизведение элементарных потоков в рамках программы.

Программный поток состоит из одного или нескольких элементарных потоков одной программы, объединенных посредством мультиплексирования. Аудио и видео элементарные потоки состоят из модулей доступа.

Данные элементарных потоков передаются в пакетах PES. Пакет PES состоит из заголовка пакета PES, за которым следуют данные пакета. Пакеты PES вводятся в блок-пакеты Программного потока.

Заголовок пакета PES начинается с 32-битового стартового кода, который идентифицирует также и поток (см. таблицу 2-22), которому принадлежат данные этого пакета. Заголовок пакета PES может содержать только метку времени воспроизведения (PTS) или две метки времени – воспроизведения и декодирования (DTS). Заголовок пакета PES содержит также и другие дополнительные поля. Данные пакета содержат переменное количество непрерывных байтов одного элементарного потока.

Пакеты PES в Программном потоке организуются в виде более крупных блок-пакетов. Такой блок-пакет начинается с заголовка блок-пакета, после которого следует ноль или несколько пакетов PES. Заголовок блок-пакета начинается с 32-битового стартового кода. Заголовок блок-пакета используется также для хранения информации синхронизации и информации о скорости передачи.

Программный поток начинается с системного заголовка, который может быть повторен. В системном заголовке передаются сведения о системных параметрах, определенные в данном потоке.

В настоящей Рекомендации | Международном стандарте не определяются кодированные данные, которые могут использоваться как часть системы условного доступа. В настоящей Рекомендации | Международном стандарте, однако, описываются механизмы для поставщиков программ, необходимые для транспортировки и идентификации этих данных в ходе обработки на этапе декодирования, и для правильного обозначения данных, которые определяются здесь.

2.5.2 Декодер конечной системы Программного потока

Семантика Программного потока и ограничения, накладываемые на эту семантику, требуют точного определения событий декодирования и моментов времени, в которые эти события происходят. Все требуемые определения сформулированы в настоящей Спецификации с использованием гипотетического декодера, известного, как Декодер конечной системы Программного потока (P-STD).

Декодер конечной системы Программного потока (P-STD) – это концептуальная модель, используемая для точного определения этих терминов и для моделирования процесса декодирования во время создания Программных потоков. Декодер P-STD определяется только для этих целей. Ни архитектура декодера P-STD, ни описанный процесс синхронизации не препятствует непрерывному, синхронному воспроизведению Программных потоков на выходе разнообразных декодеров, имеющих различные архитектуры или правила синхронизации.

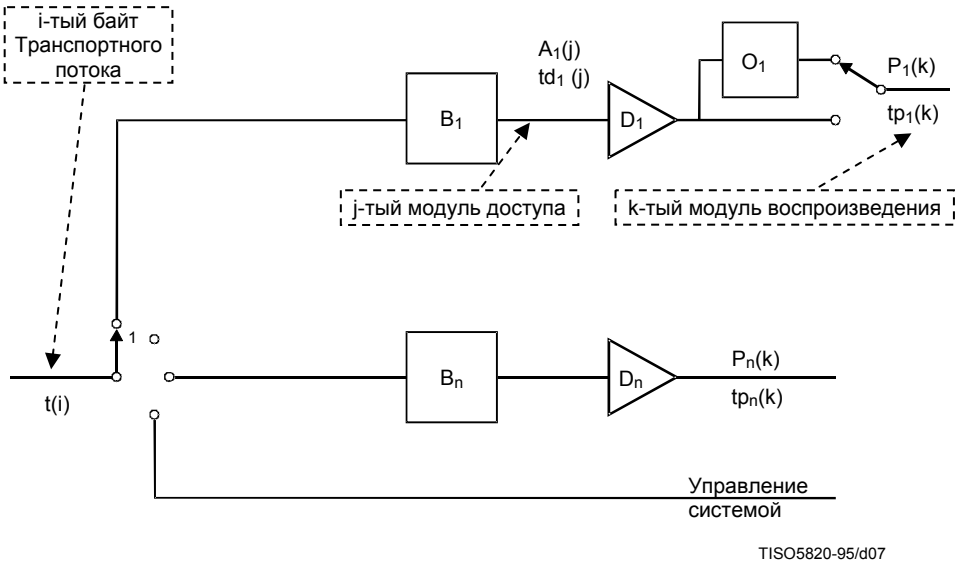


Рисунок 2-2 – Нотация декодера конечной системы Программного потока

Нижеприведенная нотация используется для описания декодера конечной системы Программного потока и частично проиллюстрирована на рисунке 2-2.

- i, i′ индексы байтов в Программном потоке. Первый байт имеет индекс 0.
- j индекс модулей доступа в элементарных потоках.
- k, k′, k″ индексы модулей представления в элементарных потоках.
- n индекс элементарных потоков.
- t(i) указывает время в секундах, в которое i-тый байт Программного потока поступает на декодер конечной системы. Значение t(0) – произвольная постоянная.
- SCR(i) время, закодированное в поле SCR, измеренное в единицах периодов системных часов, работающих с частотой 27 МГц, где i – индекс байта последнего байта поля program_clock_reference_base.
- $A_n(j)$ j-тый модуль доступа в элементарном потоке "n". Модули $A_n(j)$ нумеруются в порядке декодирования.
- $td_n(j)$ измеренное в секундах время декодирования j-того модуля доступа в элементарном потоке "n" декодером конечной системы.
- $P_n(k)$ k-тый модуль представления в элементарном потоке "n". $P_n(k)$ нумеруются в порядке в воспроизведения.
- $tp_n(k)$ измеренное в секундах время воспроизведения k-того модуля представления в элементарном потоке "n" на декодере конечной системы.
- t время, измеренное в секундах.
- $F_n(t)$ измеренная в байтах степень наполнения входного буфера декодера конечной системы для элементарного потока "n" в момент времени t.
- B_n входной буфер декодера конечной системы для элементарного потока "n".
- BS_n размер входного буфера декодера конечной системы, измеренный в байтах, для элементарного потока "n".
- D_n декодер для элементарного потока "n".
- O_n буфер переупорядочивания для элементарного видео потока "n".

2.5.2.1 Частота системных часов

Информация синхронизации, используемая в P-STD в качестве опорной, передается в нескольких полях данных, определенных в настоящей Спецификации. Эти поля определяются в § 2.5.3.3 и 2.4.3.6. Эта информация кодируется в виде отсчетов с частотой системных часов.

Значение частоты системных часов измеряется в Герцах и должно удовлетворять следующим ограничениям:

– 27 000 000 – 810 <= system_clock_frequency <= 27 000 000 + 810;

– Скорость изменения частоты системных часов system_clock_frequency во времени $\leq 75 \times 10^{-3}$ Гц/с

Нотация "system_clock_frequency" используется в нескольких местах настоящей Спецификации для указания частоты часов, соответствующей этим требованиям. Для удобства обозначения, уравнения, в которых встречаются значения PCR, PTS или DTS, приводят к получению значений времени, которые выражены с точностью до некоторого целого числа секунд, кратного (300×2^{33}/system_clock_frequency). Это происходит в результате кодирования информации синхронизации PCR в виде 33 битов с частотой дискретизации = 1/300 от частоты системных часов плюс 9 битов на остаток, и кодирование в виде 33 битов с частотой системных часов, поделенной на 300 для PTS и DTS.

2.5.2.2 Входной сигнал для декодера конечной системы Программного потока

Данные из Программного потока поступают на вход декодера конечной системы. i-тый байт поступает на его вход в момент времени t(i). Момент времени, в который этот байт поступает на вход декодера конечной системы, может быть определен из данных входного потока путем декодирования полей эталонного времени системы (SCR) и поля program_mux_rate, закодированного в заголовке блок-пакета. Поле SCR, определенное в уравнении 2-18, кодируется в виде двух частей: одной в единицах периода частоты, равной 1/300 от частоты системных часов, которая называется system_clock_reference_base (см. уравнение 2-19), и одной в единицах частоты системных часов, которая называется уравнением system_clock_reference_ext (см. уравнение 2-20). В дальнейшем значения, закодированные в этих полях, обозначаются как: SCR_base(i) и SCR_ext(i). Значение, закодированное в поле SCR, указывает время t(i), где i указывает байт, содержащий последний бит поля system_clock_reference_base.

В частности:

$$SCR(i) = SCR_base(i) \times 300 + SCR_ext(i)\,, \tag{2-18}$$

где:

$$SCR_base(i) = ((system_clock_frequency \times t(i))DIV\ 300)\%2^{33} \tag{2-19}$$

$$SCR_ext(i) = ((system_clock_frequency \times t(i))DIV\ 1)\%300 \tag{2-20}$$

Для всех других байтов время прихода входного сигнала t(i), показанное в уравнении 2-21, должно быть вычислено из значения SCR(i) и скорости, с которой прибывают данные, где скорость прибытия в каждом блок-пакете представляет собой значение, записанное в поле program_mux_rate.

$$t(i) = \frac{SCR(i')}{system_clock_frequency} + \frac{i - i'}{program_mux_rate \times 50}\,, \tag{2-21}$$

где:

- i′ индекс байта, содержащего последний бит поля system_clock_reference_base в заголовке блок-пакета
- i индекс любого байта в блок-пакете, включая заголовок блок-пакета
- SCR(i′) время. в единицах системных часов, закодированное в полях эталонной временной базы программы и полях расширения эталонной временной базы
- program_mux_rate это поле, определенное в § 2.5.3.3.

После доставки последнего байта блок-пакета, может существовать период времени, в течение которого на вход P-STD не доставляется ни одного байта.

2.5.2.3 Буферизация

Пакет данных PES из элементарного потока "n" передается на вход буфера B_n для потока "n". Передача i-того байта с входа декодера конечной системы в буфер B_n выполняется мгновенно, таким образом, этот i-тый байт поступает в буфер для потока "n", имеющий размер BS_n, в момент времени t(i).

Байты, присутствующие в заголовке блок-пакета, системных заголовках, картах Программного потока, директориях Программного потока или в заголовках пакетов PES Программного потока, таких как SCR, DTS, PTS, и полях packet_length, не доставляются ни в один из буферов, но могут использоваться для управления системой.

Размеры входных буферов BS_1 – BS_n определяются параметром размера буфера декодера P-STD в синтаксисе уравнений 2-16 и 2-17.

В момент времени декодирования $td_n(j)$, все данные модуля доступа ($A_n(j)$), который на момент времени $td_n(j)$ находится в буфере дольше всех остальных, и все непосредственно предшествующие ему байты заполнения, которые находятся в буфере в момент времени $td_n(j)$, немедленно удаляются. Время декодирования $td_n(j)$ определяется в полях DTS или PTS. Моменты времени декодирования $td_n(j + 1)$, $td_n(j + 2)$, ... модулей доступа, не содержащих кодированных полей DTS или PTS, которые расположены непосредственно после модуля доступа j, могут быть определены из информации элементарного потока. Обратитесь к Приложению C Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, ИСО/МЭК 13818-3, ИСО/МЭК 11172-2 или ИСО/МЭК 11172-3. Кроме того, см. § 2.7.5. Как только модуль доступа удаляется из буфера, он немедленно декодируется в модуль представления.

Программный поток должен быть создан в момент времени t(i), который должен выбираться так, чтобы входные буферы размера от BS_1 до BS_n в декодере конечной системы Программного потока не переполнялись и не оказывались недозаполненными. То есть, для всех значений t и n:

$$0 \le F_n(t) \le BS_n$$

и:

$$F_n(t) = 0$$

Непосредственно перед моментом времени t = t(0).

$F_n(t)$ – степень заполнения буфера B_n декодера P-STD в данный момент времени.

Исключением из этого условия является ситуация, в которой буфер B_n декодера P-STD может быть недозаполнен в том случае, когда флаг low_delay в заголовке видеопоследовательности установлен в '1' (см. § 2.4.2.6) или когда состояние trick_mode = "true" (см. § 2.4.3.8).

Для всех Программных потоков задержка, вносимая входной буферизацией декодера конечной системы, должна быть меньше или равна одной секунде, за исключением случаев для видеоданных неподвижных изображений и потоков, соответствующих ИСО/МЭК 14496. Задержка, вносимая входной буферизацией, представляет собой разницу между временем поступления байта на входной буфер и временем его декодирования.

В частности: для данных, не являющихся видеоданными неподвижных изображений или данными потоков, соответствующих ИСО/МЭК 14496, задержка ограничивается величиной:

$$tdn(j) - t(i) <= 1 \text{ с}$$

в случае видеоданных неподвижных изображений задержка ограничивается величиной:

$$tdn(j) - t(i) <= 60 \text{ с}$$

в случае потоков, соответствующих ИСО/МЭК 14496, задержка ограничивается величиной:

$$tdn(j) - t(i) <= 10 \text{ с}$$

для всех байтов, содержащихся в j –том модуле доступа.

Для Программных потоков все байты каждого блок-пакета должны поступать на декодер P-STD до прихода какого-либо байта следующего блок-пакета.

В том случае, когда флаг **low_delay** в расширении видеопоследовательности установлен в '1' (см. § 6.2.2.3 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2), буфер VBV может быть недозаполнен. В таком случае, когда рассматривается буфер Bn декодера P-STD элементарного потока в момент времени, указанный значением $td_n(j)$, в буфере B_n может не быть всех данных для модуля доступа. Когда такое случается, буфер должен повторно проверяться с интервалами, равными двум периодам поля, до тех пор, пока в буфере не будут представлены полные данные для модуля доступа. В этот момент времени модуль доступа должен быть сразу целиком удален из буфера B_n.

Допускается, чтобы буфер VBV постоянно недозаполнялся безо всяких ограничений. Декодер P-STD должен удалять данные модуля доступа из буфера B_n сразу, как только это допустимо в соответствии с вышеприведенным параграфом и любыми значениями DTS или PTS, закодированными в двоичном потоке. Декодер может не иметь возможности восстановить корректные значения времени декодирования и воспроизведения, указанные в DTS и PTS до тех пор, пока не завершится ситуация с недозаполнением буфера VBV, и в двоичном потоке не будут найдено значение PTS или DTS.

2.5.2.4 Потоки PES

Существует возможность сформировать поток данных в виде непрерывного потока пакетов PES, каждый из которых содержит один и тот же элементарный поток и имеет один и тот же идентификатор stream_id. Такой поток называется потоком PES. Модель декодера PES-STD для потока PES идентична такой модели для Программного потока, с тем исключением, что вместо SCR используется эталонное время элементарного

потока (ESCR), и вместо program_mux_rate используется ES_rate. Демультиплексор передает данные только в один буфер элементарного потока.

Размеры буфера BS_n в модели PES-STD определяются следующим образом:

– *Для видеопотока, соответствующего Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2*:

$$BS_n = VBV_{max}[profile, level] + BS_{oh}$$

BS_{oh} = (1/750) секунд × R_{max}[profile, level], где VBV_{max}[profile, level] и R_{max}[profile, level] представляют собой максимальный размер VBV и скорость передачи для каждого профиля, уровня и слоя, как определено в таблицах 8-14 и 8-13, соответственно, Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2. Значения BS_{oh} распределены для заголовка пакета PES.

– *Для видеопотока, соответствующего ИСО/МЭК 11172-2*:

$$BS_n = VBV_{max} + BS_{oh}$$

BS_{oh} = (1/750) секунд × R_{max}, где R_{max} и vbv_max обозначают максимальную скорость передачи и максимальное значение vbv_buffer_size для двоичного потока с ограниченными параметрами в ИСО/МЭК 11172-2, соответственно.

– *Для аудиопотока, соответствующего ИСО/МЭК 11172-3 или ИСО/МЭК 13818-3*:

$$BS_n = 2848 \text{ байтов}$$

– *Для видеопотока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10*:

$$BS_n = 1200 \times MaxCPB[level] + BS_{oh}$$

где MaxCPB[level] для каждого уровня определяется в таблице A.1 (Ограничения уровня) Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10.

2.5.2.5 Декодирование и воспроизведение

Декодирование и воспроизведение в декодере конечной системы Программного потока выполняются так же, как определено для декодера конечной системы Транспортного потока в § 2.4.2.4 и § 2.4.2.5, соответственно.

2.5.2.6 Расширения декодера P-STD для передачи данных, соответствующих ИСО/МЭК 14496

Для декодирования данных, соответствующих ИСО/МЭК 14496, передаваемых в Программном потоке, модель декодера P-STD расширяется. Декодирование в декодере P-STD отдельных элементарных потоков, соответствующих ИСО/МЭК 14496, описано в § 2.11.2. В § 2.11.3 определены расширения декодера P-STD и параметры для декодирования видеокадров, соответствующих ИСО/МЭК 14496, и связанных с ними потоков.

2.5.2.7 Расширения декодера P-STD для потоков, в которых передаются видеоданные, соответствующие Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10

Декодирование видеопотоков, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, и передаваемых в Программном потоке в рамках модели P-STD, описано в § 2.14.3.2.

2.5.3 Спецификация синтаксиса и семантики Программного потока

Поток байтов описывается при помощи следующего синтаксиса.

2.5.3.1 Программный поток

См. таблицу 2-37.

Таблица 2-37 – Программный поток

Синтаксис	Количество битов	Мнемоника
MPEG2_program_stream() {		
do {		
pack()		
} while (nextbits() = = pack_start_code)		
MPEG_program_end_code	**32**	**bslbf**
}		

2.5.3.2 Семантические определения полей в Программном потоке

MPEG_program_end_code – MPEG_program_end_code – это строка битов '0000 0000 0000 0000 0000 0001 1011 1001' (0x000001B9). Она завершает Программный поток.

2.5.3.3 Уровень блок-пакетов Программного потока

См. таблицы 2-38 и 2-39.

Таблица 2-38 – Блок-пакет Программного потока

Синтаксис	Количество битов	Мнемоника
pack() {		
pack_header()		
while (nextbits() = -= packet_start_code_prefix) {		
PES_packet()		
}		
}		

Таблица 2-39 – Заголовок блок-пакета Программного потока

Синтаксис	Количество битов	Мнемоника
pack_header() {		
pack_start_code	**32**	**bslbf**
'01'	**2**	**bslbf**
system_clock_reference_base [32..30]	**3**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_base [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_base [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
system_clock_reference_extension	**9**	**uimsbf**
marker_bit	**1**	**bslbf**
program_mux_rate	**22**	**uimsbf**
marker_bit	**1**	**bslbf**
marker_bit	**1**	**bslbf**
зарезервировано	**5**	**bslbf**
pack_stuffing_length	**3**	**uimsbf**
for (i = 0; i < pack_stuffing_length; i++) {		
stuffing_byte	**8**	**bslbf**
}		
if (nextbits() = = system_header_start_code) {		
system_header ()		
}		
}		

2.5.3.4 Семантические определения полей в блок-пакете Программного потока

pack_start_code – Код pack_start_code – это строка битов '0000 0000 0000 0000 0000 0001 1011 1010' (0x000001BA). Он определяет начало блок-пакета.

system_clock_reference_base; system_clock_reference_extension – Эталонное время системы (SCR) – это 42-битовое поле, закодированное в виде двух частей. Первая часть (system_clock_reference_base) – это 33-битовое поле, значение которого определено величиной SCR_base(i), как показано в уравнении 2-19. Вторая часть (system_clock_reference_extension) – это 9-битовое поле, значение которого определено величиной SCR_ext(i), как показано в уравнении 2-20. Значение SCR указывает предполагаемое время прибытия байта, содержащего последний бит поля system_clock_reference_base, на вход конечного декодера программы.

Требования к частоте кодирования для поля SCR приводятся в § 2.7.1.

marker_bit – Поле marker_bit – это 1-битовое поле, которое имеет значение '1'.

program_mux_rate – Это 22-битовое целое число, определяющее скорость, с которой декодер P-STD принимает Программный поток в течение времени блок-пакета, который его содержит. Величина числа program_mux_rate указывается в единицах по 50 байтов в секунду. Значение '0' является запрещенным.

Значение, записанное в program_mux_rate, используется в § 2.5.2 для определения времени прибытия байтов на вход декодера P-STD. Значение, закодированное в поле program_mux_rate, может быть различным для различных блок-пакетов мультиплексированного программного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

pack_stuffing_length – Это 3-битовое целое число, определяющее число байтов заполнения, которые следуют после этого поля.

stuffing_byte – Это фиксированное 8-битовое значение, равное '1111 1111', которое может вводиться кодером, например, для того чтобы выполнить требования канала передачи. Оно отбрасывается декодером. В каждом заголовке блок-пакета должно присутствовать не более 7 байтов заполнения.

2.5.3.5 Системный заголовок

См. таблицу 2-40.

Таблица 2-40 – Системный заголовок Программного потока

Синтаксис	Количество битов	Мнемоника
system_header () {		
system_header_start_code	**32**	**bslbf**
header_length	**16**	**uimsbf**
marker_bit	**1**	**bslbf**
rate_bound	**22**	**uimsbf**
marker_bit	**1**	**bslbf**
audio_bound	**6**	**uimsbf**
fixed_flag	**1**	**bslbf**
CSPS_flag	**1**	**bslbf**
system_audio_lock_flag	**1**	**bslbf**
system_video_lock_flag	**1**	**bslbf**
marker_bit	**1**	**bslbf**
video_bound	**5**	**uimsbf**
packet_rate_restriction_flag	**1**	**bslbf**
reserved_bits	**7**	**bslbf**
while (nextbits () = = '1') {		
stream_id	**8**	**uimsbf**
'11'	**2**	**bslbf**
P-STD_buffer_bound_scale	**1**	**bslbf**
P-STD_buffer_size_bound	**13**	**uimsbf**
}		
}		

2.5.3.6 Семантические определения полей в системном заголовке

system_header_start_code – Код system_header_start_code – это строка битов '0000 0000 0000 0000 0000 0001 1011 1011' (0x000001BB). Он определяет начало системного заголовка.

header_length – Это 16-битовое поле указывает длину системного заголовка в байтах, которые следуют после поля header_length. В дальнейших расширениях настоящей Спецификации системный заголовок может быть расширен.

rate_bound – 22-битовое поле. Значение rate_bound – это целочисленная величина, которая больше или равна максимальному значению поля program_mux_rate, закодированному в любом блок-пакете Программного потока. Оно может использоваться декодером для оценки того, способен ли он декодировать целый поток.

audio_bound – 6-битовое поле. Audio_bound – это целочисленная величина, значение которой лежит в пределах от 0 до 32 включительно, и установлено большим или равным максимальному количеству передаваемых в Программном потоке аудиопотоков, соответствующих ИСО/МЭК 13818-3 и ИСО/МЭК 11172-3, для которых процессы декодирования являются активными одновременно. В настоящем подразделе процесс декодирования аудиопотока, соответствующего ИСО/МЭК 13818-3 и ИСО/МЭК 11172-3, является активным, если буфер STD не пуст, или если в модели декодера P-STD присутствует модуль представления.

fixed_flag – Флаг fixed_flag – это 1-битовый флаг. Когда он установлен в '1', он указывает, что передача ведется с постоянной скоростью. Когда он установлен в '0', он указывает, что передача ведется с переменной скоростью и. Во время работы с постоянной скоростью передачи, значения, закодированные во всех полях system_clock_reference мультиплексированного потока, соответствующего Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1, должны подчиняться следующему линейному уравнению:

$$SCR_base(i) = ((c1 \times i + c2)\ DIV\ 300)\ \%\ 2^{33} \tag{2-22}$$

$$SCR_ext(i) = ((c1 \times i + c2)\ DIV\ 300)\ \%\ 300 \tag{2-23}$$

где:

c1 не содержащая комплексных величин постоянная, действительная для всех i.

c2 не содержащая комплексных величин постоянная, действительная для всех i.

i индекс байта в мультиплексированном потоке, соответствующем Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1, содержащего последний бит любого поля system_clock_reference в этом потоке.

CSPS_flag – Флаг CSPS_flag – это 1-битовое поле. Если его значение установлено в '1', то Программный поток удовлетворяет ограничениям, определенным в § 2.7.9.

system_audio_lock_flag – Флаг system_audio_lock_flag – это 1-битовое поле, указывающее, что в декодере конечной системы существует определенное постоянное рациональное соотношение между частотой дискретизации звукового сигнала и частотой системных часов (system_clock_frequency). Значение system_clock_frequency определяется в § 2.5.2.1, а частота дискретизации звукового сигнала определяется в стандарте ИСО/МЭК 13818-3. Флаг system_audio_lock_flag может быть установлен в '1' тогда, когда для всех модулей представления во всех элементарных аудиопотоках в Программном потоке, отношение значения system_clock_frequency к реальной частоте дискретизации звукового сигнала (SCASR) является постоянным и равным величине, указанной в следующей таблице для номинальной частоты дискретизации звукового сигнала, указанной в аудиопотоке.

$$SCASR = \frac{system_clock_frequency}{audio_sample_rate_in_the_P-STD} \tag{2-24}$$

Нотация $\frac{X}{Y}$ обозначает деление, содержащее комплексных величин.

Номинальная частота дискретизации звукового сигнала (кГц)	16	32	22.05	44.1	24	48
SCASR	27 000 000 / 16 000	27 000 000 / 32 000	27 000 000 / 22 050	27 000 000 / 44 100	27 000 000 / 24 000	27 000 000 / 48 000

system_video_lock_flag – Поле system_video_lock_flag – это 1-битовое поле, показывающее, что в конечном системном декодере существует определенное постоянное рациональное соотношение между временной базой видеосигнала и частотой системных часов. Поле system_video_lock_flag может быть установлено в '1' только тогда, когда для всех модулей представления во всех элементарных потоках программ, соответствующих Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1, отношение параметра system_clock_frequency к фактической частоте временной базы видеосигнала постоянно.

Для видеопотоков, соответствующих ИСО/МЭК 11172-2 и Рекомендации МСЭ-Т Н.262 | ИСО/МЭК 13818-2, если флаг system_video_lock_flag установлен в '1', отношение параметра system_clock_frequency к фактической частоте кадров (SCFR) должно быть постоянным и равным значению, показанному в следующей таблице при номинальной частоте кадров, отображаемой в видеопотоке.

Для видеопотоков, соответствующих ИСО/МЭК 14496-2, если флаг system_video_lock_flag установлен в '1', то временная база видеопотока ИСО/МЭК 14496-2, как определено в параметре vop_time_increment_resolution, должна быть жестко связана с STC и точно равна N умноженному на значение параметра system_clock_frequency, деленным на K, где N и K – целые числа, имеющие фиксированные значения в пределах каждой последовательности визуальных объектов, причем K больше или равно N.

Для видеопотоков, соответствующих Рекомендации МСЭ-Т Н.264 | ИСО/МЭК 14496-10, частота временной базы AVC определяется параметром AVC time_scale. Если в видеопотоке AVC флаг system_video_lock_flag установлен в '1', то частота временной базы AVC должна быть синхронизирована с STC и точно равна N, умноженному на значение параметра system_clock_frequency, поделенному на K, где N и K – целые числа,

имеющие фиксированное значение в пределах каждой видеопоследовательности AVC, причем K больше или равно N.

$$SCFR = \frac{system_clock_frequency}{frame_rate_in_the_P-STD} \tag{2-25}$$

Номинальная частота кадров (Гц)	23.976	24	25	29.97	30	50	59.94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

Значения отношения SCFR являются точными. Реальная частота кадров слегка отличается от номинальной частоты в тех случаях, когда номинальная частота равна 23.976, 29.97 или 59.94 кадров в секунду.

video_bound – Поле video_bound – это 5-битовое целое число в диапазоне от 0 до 16 включительно, установленное в значение, большее или равное максимальному количеству передаваемых в Программном потоке видеопотоков, в которых процессы декодирования одновременно активны. В данном подпункте считается, что процесс декодирования видеопотока активен, если один из буферов модели декодера P-STD не пуст или если в модели декодера P-STD присутствует модуль представления.

packet_rate_restriction_flag – Флаг packet_rate_restriction_flag – это 1-битовый флаг. Если флаг CSPS установлен в '1', то флаг packet_rate_restriction_flag указывает, какое определенное в § 2.7.9 ограничение применимо для скорости передачи пакетов. Если флаг CSPS установлен в значение '0', то значение флага packet_rate_restriction_flag не определено.

reserved_bits – Это 7-битовое поле зарезервировано организациями ИСО/МЭК для будущего использования. До тех пор, пока в документах МСЭ-Т | ИСО/МЭК не будет определено иного, оно должно иметь значение '111 1111'.

stream_id – Идентификатор потока stream_id – это 8-битовое поле, которое указывает кодировку и номер элементарного потока в потоке, к которому относятся следующие после него поля P-STD_buffer_bound_scale и P-STD_buffer_size_bound.

Если stream_id равен '1011 1000', то следующие после stream_id поля P-STD_buffer_bound_scale и P-STD_buffer_size_bound относятся ко всем аудиопотокам в Программном потоке.

Если stream_id равен '1011 1001', то следующие после stream_id поля P-STD_buffer_bound_scale и P-STD_buffer_size_bound относятся ко всем видеопотокам в Программном потоке.

Если stream_id принимает любое другое значение, то этот байт должен иметь значение, большее или равное '1011 1100' и его следует понимать, как указание кодировки номера элементарного потока в соответствии с таблицей 2-22.

Каждый элементарный поток, присутствующий в Программном потоке, должен иметь свои значения P-STD_buffer_bound_scale и P-STD_buffer_size_bound, которые однократно определены при помощи этого механизма в каждом системного заголовке.

P-STD_buffer_bound_scale – Поле P-STD_buffer_bound_scale – это 1-битовое поле, которое указывает коэффициент масштабирования, использованный для понимания следующего за ним поля P-STD_buffer_size_bound. Если предшествующий ему индикатор stream_id указывает аудиопоток, то поле P-STD_buffer_bound_scale должно иметь значение '0'. Если предшествующий ему индикатор stream_id указывает видеопоток, то поле P-STD_buffer_bound_scale должно иметь значение '1'. Для всех других типов потока, значение поля P-STD_buffer_bound_scale может быть или ‘1’, или ‘0’.

P-STD_buffer_size_bound – Поле P-STD_buffer_size_bound – это 13-битовое целое число без знака, определяющее значение большее или равное максимальному размеру входного буфера декодера P-STD (BS_n) для всех пакетов потока "n" в Программном потоке. Если поле P-STD_buffer_bound_scale имеет значение '0', то P-STD_buffer_size_bound измеряет границу размера буфера в единицах по 128 байтов. Если поле P-STD_buffer_bound_scale имеет значение '1', то P-STD_buffer_size_bound измеряет границу размера буфера в единицах по 1024 байта. Таким образом, если:

$$(P-STD_buffer_bound_scale == 0)$$
$$BS_n \le P-STD_buffer_size_bound \times 128$$

в ином случае:

$$BS_n \le P-STD_buffer_size_bound \times 1024$$

2.5.3.7 Уровень пакетов Программного потока

Уровень пакетов Программного потока определяется уровнем пакетов PES в § 2.4.3.6.

2.5.4 Карта Программного потока

Карта Программного потока (PSM) содержит описание элементарных потоков в Программном потоке и описание их соответствия друг другу. Эта структура не должна изменяться при передаче в Транспортном потоке. Карта PSM присутствует в виде пакета PES, когда значение stream_id = 0xBC (см. таблицу 2-22).

ПРИМЕЧАНИЕ. – Этот синтаксис отличается от синтаксиса пакета PES, описанного в § 2.4.3.6.

Поля descriptor() определены в § 2.6.

2.5.4.1 Синтаксис карты Программного потока

См. таблицу 2-41.

Таблица 2-41 – Карта Программного потока

Синтаксис	Количество битов	Мнемоника
program_stream_map() {		
packet_start_code_prefix	**24**	**bslbf**
map_stream_id	**8**	**uimsbf**
program_stream_map_length	**16**	**uimsbf**
current_next_indicator	**1**	**bslbf**
зарезервировано	**2**	**bslbf**
program_stream_map_version	**5**	**uimsbf**
зарезервировано	**7**	**bslbf**
marker_bit	**1**	**bslbf**
program_stream_info_length	**16**	**uimsbf**
for (i = 0; i < N; i++) {		
descriptor()		
}		
elementary_stream_map_length	**16**	**uimsbf**
for (i = 0; i < N1; i++) {		
stream_type	**8**	**uimsbf**
elementary_stream_id	**8**	**uimsbf**
elementary_stream_info_length	**16**	**uimsbf**
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	**32**	**rpchof**
}		

2.5.4.2 Семантические определения полей в карте Программного потока

packet_start_code_prefix – Префикс packet_start_code_prefix – это 24-битовый код. Вместе со следующим после него идентификатором map_stream_id, он образует стартовый код пакета, который идентифицирует начало пакета. Префикс packet_start_code_prefix – это строка битов '0000 0000 0000 0000 0000 0001' (в шестнадцатеричном счислении 0x000001).

map_stream_id – Это 8-битовое поле, значение которого должно быть 0xBC.

program_stream_map_length – Поле program_stream_map_length – это 16-битовое поле, указывающее общее число байтов в поле program_stream_map, которое следует сразу же после этого поля. Максимальное значение этого поля составляет 1018 (0x3FA).

current_next_indicator – Это 1-битовое поле, которое, когда оно установлено в '1', указывает, что в настоящее время может быть применена переданная карта Программного потока. Когда этот бит установлен в ‘0’, он указывает, что переданная карта Программного потока пока еще применена быть не может, и должна быть следующей таблицей, которая станет действительной.

program_stream_map_version – Это 5-битовое поле представляет собой номер версии полной карты Программного потока. Номер версии должен увеличиваться на 1 (по модулю 32) при каждом изменении определения карты Программного потока. Когда индикатор current_next_indicator установлен в '1', то поле program_stream_map_version должно соответствовать применяемой в настоящее время карте Программного потока. Когда индикатор current_next_indicator установлен в '0', то поле program_stream_map_version должно соответствовать карте Программного потока, которая будет применима следующей.

program_stream_info_length – Поле program_stream_info_length представляет собой 16-битовое поле, указывающее общую длину дескрипторов, который следуют сразу же после этого поля.

marker_bit – Поле marker_bit – это 1-битовое поле, которое имеет значение '1'.

elementary_stream_map_length – Это 16-битовое поле, определяющее общую длину, в байтах, всей информации элементарного потока в этой карте Программного потока. Оно включает в себя поля stream_type, elementary_stream_id и elementary_stream_info_length.

stream_type – Это 8-битовое поле определяет тип потока в соответствии с таблицей 2-34. Поле stream_type должно определять только элементарные потоки, содержащиеся в пакетах PES. Значение 0x05 запрещено.

elementary_stream_id – Поле elementary_stream_id – это 8-битовое поле, указывающее значение поля stream_id в заголовках пакетов PES, в которых сохраняется этот элементарный поток.

elementary_stream_info_length – Поле elementary_stream_info_length – это 16-битовое поле, указывающее длину (в байтах) дескрипторов, который следуют сразу же после этого поля.

CRC_32 – Это 32-битовое поле, которое содержит значение кода CRC, которое в результате приводит к получению нулей на выходе регистров в декодере, определенного в Приложении A, после обработки полной секции карты программного потока.

2.5.5 Каталог Программного потока

Каталог для потока в целом формируется из всех данных каталога, передаваемых в пакетах Каталога Программного потока, идентифицированных при помощи идентификатора directory_stream_id. Синтаксис для пакетов program_stream_directory определяется в таблице 2-42.

ПРИМЕЧАНИЕ 1. – Этот синтаксис отличается от синтаксиса пакета PES, описанного в § 2.4.3.6.

Записи в каталоге могут потребоваться для ссылки на изображение I в видеопотоке, как определено в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 и ИСО/МЭК 11172-2. Если изображению I, которое указано в записи каталога, предшествует заголовок последовательности, в котором нет вложенных заголовков кадров изображения, то запись каталога должна указывать первый байт заголовка последовательности. Если изображению I, которое указано в записи каталога, предшествует заголовок группы кадров изображения, в котором нет вложенных заголовков кадров изображения, и нет заголовка последовательности, который непосредственно предшествует этому изображению, запись каталога должна указывать первый байт заголовка группы кадров изображения. Любое другое изображение, которое указано в записи каталога, должно указываться первым байтом заголовка кадра изображения.

ПРИМЕЧАНИЕ 2. – Рекомендуется, чтобы изображения I, которые следуют непосредственно после заголовка последовательности, указывались в структурах каталога таким образом, чтобы каталог содержал запись для каждой точки, в которой декодер может быть полностью переустановлен.

Записи в каталоге могут потребоваться для ссылки на изображение IDR или изображения, связанные с точкой восстановления сообщения SEI в видеопотоке AVC. Каждая такая запись в каталоге указывает на первый байт модуля доступа AVC.

Ссылки в каталоге на аудиопотоки, определенные в ИСО/МЭК 13818-3 и ИСО/МЭК 11172-3, должны быть синхрословами звукового кадра.

ПРИМЕЧАНИЕ 3. – Рекомендуется, чтобы расстояние между модулями доступа, на которые имеются ссылки, не превышала половины секунды.

Модули доступа должны указываться в пакете program_stream_directory в том же порядке, в котором они присутствуют в двоичном потоке.

2.5.5.1 Синтаксис пакета каталога Программного потока

См. таблицу 2-42.

Таблица 2-42 – Пакет каталога Программного потока

Синтаксис	Количество битов	Мнемоника
directory_PES_packet(){		
packet_start_code_prefix	**24**	**bslbf**
directory_stream_id	**8**	**uimsbf**
PES_packet_length	**16**	**uimsbf**
number_of_access_units	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[44..30]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
prev_directory_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[44..30]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
next_directory_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
for (i = 0; i < number_of_access_units; i++) {		
packet_stream_id	**8**	**uimsbf**
PES_header_position_offset_sign	**1**	**tcimsbf**
PES_header_position_offset[43..30]	**14**	**uimsbf**
marker_bit	**1**	**bslbf**
PES_header_position_offset[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
PES_header_position_offset[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
reference_offset	**16**	**uimsbf**
marker_bit	**1**	**bslbf**
зарезервировано	**3**	**bslbf**
PTS[32..30]	**3**	**uimsbf**
marker_bit	**1**	**bslbf**
PTS[29..15]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
PTS[14..0]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
bytes_to_read[22..8]	**15**	**uimsbf**
marker_bit	**1**	**bslbf**
bytes_to_read[7..0]	**8**	**uimsbf**
marker_bit	**1**	**bslbf**
intra_coded_indicator	**1**	**bslbf**
coding_parameters_indicator	**2**	**bslbf**
зарезервировано	**4**	**bslbf**
}		
}		

2.5.5.2 Семантические определения полей в Справочнике Программного потока

packet_start_code_prefix – Префикс packet_start_code_prefix – это 24-битовый код. Вместе с идентификатором stream_id, который следует после него, он образует стартовый код пакета, который обозначает начало пакета. Префикс packet_start_code_prefix – это строка битов '0000 0000 0000 0000 0000 0001' (0x000001 в шестнадцатеричной системе счисления).

directory_stream_id – Это 8-битовое поле должно иметь значение '1111 1111' (0xFF).

PES_packet_length – Поле PES_packet_length – это 16-битовое поле, указывающее общее количество байтов в каталоге program_stream_directory, который следует сразу после этого поля (см. таблицу 2-22).

number_of_access_units – Это 15-битовое поле представляет собой количество модулей доступа (access_units), указанных в данном пакете PES каталога.

prev_directory_offset – Это 45-битовое целое число без знака, содержит сдвиг адреса первого байта стартового кода пакета предыдущего пакета каталога Программного потока. Этот сдвиг адреса определяется относительно первого байта стартового кода пакета, в котором содержится данное поле previous_directory_offset. Значение ‘0' указывает, что предыдущего пакета каталога Программного потока не существует.

next_directory_offset – Это 45-битовое целое число без знака указывает сдвиг адреса первого байта стартового кода пакета следующего пакета каталога Программного потока. Этот сдвиг адреса определяется относительно первого байта стартового кода пакета, в котором содержится данное поле next_directory_offset. Значение ‘0' указывает, что следующего пакета каталога Программного потока не существует.

packet_stream_id – Это 8-битовое поле является идентификатором потока stream_id элементарного потока, в котором содержится модуль доступа, указанный этой записью каталога.

PES_header_position_offset_sign – Это 1-битовое поле представляет собой арифметический знак описанного сдвига PES_header_position_offset, значения которого следует сразу после него. Значение '0' указывает, что сдвиг PES_header_position_offset является положительным. Значение '1' указывает, что сдвиг PES_header_position_offset является отрицательным.

PES_header_position_offset – Это 44-битовое целое число без знака представляет собой сдвиг байтового адреса первого байта пакета PES, в котором содержится указанный модуль доступа. Сдвиг адреса определяется относительно первого байта стартового кода пакета, в котором содержится это поле PES_header_position_offset. Значение ‘0' указывает, что ни один модуль доступа не указывается.

reference_offset – Это 16-битовое поле представляет собой целое число без знака, указывающее положение первого байта указанного модуля доступа, измеренное в байтах относительно первого байта пакета PES, в котором содержится первый байт указанного модуля доступа.

PTS (presentation_time_stamp) – Это 33-битовое поле представляет собой метку времени воспроизведения (PTS) модуля доступа, который указан в записи каталога. Семантика кодирования поля PTS соответствует семантике описанной в § 2.4.3.6.

bytes_to_read – Это 23-битовое целое число без знака представляет собой количество байтов в Программном потоке, расположенных после байта, указанного полем reference_offset, которые необходимы для полного декодирования модуля доступа. Это значение включает в себя все байты, мультиплексированные на системном уровне, включая те, что содержат информацию из других потоков.

intra_coded_indicator – Это 1-битовый флаг. Когда он установлен в '1', он указывает, что в модуле доступа, указанном в каталоге, кодирование с предсказанием не используется. Он не зависит от других параметров кодирования, которые могут потребоваться для декодирования модуля доступа. Например, это поле должно иметь значение '1' для видеокадров 'Intra', тогда как для кадров 'P' и 'B' этот бит должен иметь значение '0'. Для всех пакетов PES, содержащих данные потоков, не соответствующих Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, это поле не определено (см. таблицу 2-43).

Таблица 2-43 – Индикатор Intra_coded

Величина	Значение
0	Не 'Intra'
1	'Intra'

coding_parameters_indicator – Это 2-битовое поле используется для обозначения местоположения параметров кодирования, необходимых для декодирования указанных в каталоге модулей доступа. Например, это поле может использоваться для определения местоположения матриц квантования для видеокадров.

Таблица 2-44 – Индикатор coding_parameters

Величина	Значение
00	Все параметры кодирования установлены в значения "по умолчанию"
01	В данном модуле доступа установлены все параметры кодирования, как минимум, один из них не установлен в значение "по умолчанию"
10	В данном модуле доступа установлены некоторые параметры кодирования
11	В данном модуле доступа не установлен ни один из параметров кодирования

2.6 Дескрипторы программ и элементов программ

Дескрипторы программ и элементов программ представляют собой структуры, которые могут использоваться для расширения определений программ и элементов программ. Все дескрипторы имеют формат, который начинается с 8-битовой метки. После метки следует 8-битовое поле длины дескриптора и поле данных.

2.6.1 Семантические определения полей в дескрипторах программ и элементов программ

Приведенная ниже семантика применяется в дескрипторах, определенных в § 2.6.2–§ 2.6.34.

descriptor_tag – Метка descriptor_tag – это 8-битовое поле, которое идентифицирует каждый дескриптор.

В таблице 2-45 приведены значения метки дескриптора, определенные в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, зарезервированные в соответствии с Рекомендацией МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, и доступные для пользователя. Значение 'X' в столбцах TS или PS означает применимость дескриптора либо в Транспортном потоке, либо в Программном потоке, соответственно. Отметим, что значения полей в дескрипторе могут зависеть от того, в каком потоке они используются. В приведенной ниже семантике дескриптора рассмотрен каждый случай.

descriptor_length – Поле descriptor_length – это 8-битовое поле, определяющее количество байтов дескриптора, который следует сразу после поле descriptor_length.

Таблица 2-45 – Дескрипторы программ и элементов программ

descriptor_tag	TS	PS	Идентификатор
0	нет	нет	Зарезервирован
1	нет	нет	Зарезервирован
2	X	X	video_stream_descriptor
3	X	X	audio_stream_descriptor
4	X	X	hierarchy_descriptor
5	X	X	registration_descriptor
6	X	X	data_stream_alignment_descriptor
7	X	X	target_background_grid_descriptor
8	X	X	video_window_descriptor
9	X	X	CA_descriptor
10	X	X	ISO_639_language_descriptor
11	X	X	system_clock_descriptor
12	X	X	multiplex_buffer_utilization_descriptor
13	X	X	copyright_descriptor
14	X		maximum_bitrate_descriptor
15	X	X	private_data_indicator_descriptor
16	X	X	smoothing_buffer_descriptor
17	X		STD_descriptor
18	X	X	IBP_descriptor
19-26	X		Определены в ИСО/МЭК 13818-6
27	X	X	MPEG-4_video_descriptor
28	X	X	MPEG-4_audio_descriptor
29	X	X	IOD_descriptor
30	X		SL_descriptor

Таблица 2-45 – Дескрипторы программ и элементов программ

descriptor_tag	TS	PS	Идентификатор
31	X	X	FMC_descriptor
32	X	X	external_ES_ID_descriptor
33	X	X	MuxCode_descriptor
34	X	X	FmxBufferSize_descriptor
35	X		multiplexbuffer_descriptor
36	X	X	content_labeling_descriptor
37	X	X	metadata_pointer_descriptor
38	X	X	metadata_descriptor
39	X	X	metadata_STD_descriptor
40	X	X	Видеодескриптор AVC
41	X	X	IPMP_descriptor (определен в ИСО/МЭК 13818-11, MPEG-2 IPMP)
42	X	X	Дескриптор синхронизации AVC и HRD
43	X	X	MPEG-2_AAC_audio_descriptor
44	X	X	Дескриптор IPMP (определен в ИСО/МЭК 13818-11, MPEG-2 IPMP)
45-63	нет	нет	Зарезервировано в соответствии с Рек. МСЭ-Т H.222.0 | ИСО/МЭК13818-11.
64-255	n/a	n/a	Конфиденциальный дескриптор пользователя

2.6.2 Дескриптор видеопотока

Дескриптор видеопотока содержит базовую информацию, которая определяет параметры кодирования элементарного видеопотока, описанные в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 11172-2 (см. таблицу 2-46).

Таблица 2-46 – Дескриптор видеопотока

Синтаксис	Количество битов	Мнемоника
video_stream_descriptor(){		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
multiple_frame_rate_flag	**1**	**bslbf**
frame_rate_code	**4**	**uimsbf**
MPEG_1_only_flag	**1**	**bslbf**
constrained_parameter_flag	**1**	**bslbf**
still_picture_flag	**1**	**bslbf**
if (MPEG_1_only_flag = = '0'){		
profile_and_level_indication	**8**	**uimsbf**
chroma_format	**2**	**uimsbf**
frame_rate_extension_flag	**1**	**bslbf**
Зарезервированы	**5**	**bslbf**
}		
}		

2.6.3 Семантические определения полей в дескрипторе видеопотока

multiple_frame_rate_flag – Это 1-битовое поле, когда оно установлено в '1', оно указывает, что в видеопотоке могут присутствовать несколько скоростей передачи кадров. Когда оно установлено в '0', это указывает, что в видеопотоке присутствует только одна скорость передачи кадров.

frame_rate_code – Это 4-битовое поле, определенное в § 6.3.3 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, за исключением случая, когда флаг multiple_frame_rate_flag установлен в '1'. Указание определенной скорости передачи кадров позволяет также иметь в видеопотоке некоторые другие скорости передачи кадров, которые определены в таблице 2-47:

Таблица 2-47 – Код скорости передачи кадров

Код	Также включает в себя
23.976	
24.0	23.976
25.0	
29.97	23.976
30.0	23.976 24.0 29.97
50.0	25.0
59.94	23.976 29.97
60.0	23.976 24.0 29.97 30.0 59.94

MPEG_1_only_flag – Это 1-битовое поле, которое, когда установлено в '1' указывает, что видеопоток содержит только данные, соответствующие ИСО/МЭК 11172-2. Если оно установлено в '0', то видеопоток может содержать как видеоданные, соответствующие Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, так и видеоданные с ограниченными параметрами, соответствующие ИСО/МЭК 11172-2.

constrained_parameter_flag – Это 1-битовое поле, которое, когда установлено в '1' указывает, что видеопоток не должен содержать видеоданных с неограниченными параметрами, соответствующие ИСО/МЭК 11172-2. Если это поле установлено в '0', то видеопоток может содержать как видеоданные с ограниченными параметрами, так и видеоданные с неограниченными параметрами, соответствующие ИСО/МЭК 11172-2. Если флаг MPEG_1_only_flag установлен в '0', то флаг constrained_parameter_flag должен быть установлен в '1'.

still_picture_flag – Это 1-битовое поле, которое, когда оно установлено в '1' указывает, что видеопоток содержит только неподвижные изображения. Если этот бит установлен в '0', то видеопоток может содержать данные как подвижных, так и неподвижных изображений.

profile_and_level_indication – Это 8-битовое поле кодируется точно так же, как поля profile_and_level_indication видеопотока, соответствующего Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2. Значение этого поля указывает профиль и уровень, которые больше или равны любому профилю и уровню в любой последовательности в соответствующем видеопотоке. В настоящем подразделе считается, что потоком с Основным профилем на Низком уровне (Main Profile at Low Level (MP @ LL)) является поток с неограниченными параметрами, соответствующий ИСО/МЭК 11172-2.

chroma_format – Это 2-битовое поле кодируется точно так же, как поля chroma_format видеопотока, соответствующего Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2. Значение этого поля должно быть, как минимум, больше или равно значению поля chroma_format в любой видеопоследовательности соответствующего видеопотока. В настоящем подразделе считается, что видеопоток, соответствующий ИСО/МЭК 11172-2, содержит поле chroma_format, имеющее значение '01', что указывает формат 4:2:0.

frame_rate_extension_flag – Это 1-битовый флаг, который, когда установлен в '1' указывает, что в любой из видеопоследовательностей видеопотока, соответствующего Рекомендации. МСЭ-Т H.262 | ИСО/МЭК 13818-2, какое либо из полей frame_rate_extension_n и frame_rate_extension_d, или оба эти поля, не равны нулю. В настоящем подразделе считается, что в видеопотоке, соответствующем ИСО/МЭК 11172-2, введены ограничения, согласно которым оба эти поля установлены в ноль.

2.6.4 Дескриптор аудиопотока

Дескриптор аудиопотока содержит базовую информацию, которая определяет версию кодирования элементарного аудиопотока, описанную в ИСО/МЭК 13818-3 или ИСО/МЭК 11172-3 (см. таблицу 2-48).

Таблица 2-48 – Дескриптор аудиопотока

Синтаксис	Количество битов	Мнемоника
audio_stream_descriptor(){		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
free_format_flag	**1**	**bslbf**
ID	**1**	**bslbf**
layer	**2**	**bslbf**
variable_rate_audio_indicator	**1**	**bslbf**
зарезервировано	**3**	**bslbf**
}		

2.6.5 Семантические определения полей в дескрипторе аудиопотока

free_format_flag – Это 1-битовое поле, когда оно установлено в '1', указывает, что аудиопоток может содержать один или несколько аудиокадров, в которых поле bitrate_index установлено в '0000'. Если это поле установлено в '0', то ни в одном аудиокадре этого аудиопотока поле bitrate_index не имеет значения '0000' (см. § 2.4.2.3 стандарта ИСО/МЭК 13818-3).

ID – Это 1-битовое поле, когда оно установлено в '1' указывает, что в каждом аудиокадре этого аудиопотока поле ID установлено в '1' (см. § 2.4.2.3 стандарта ИСО/МЭК 13818-3).

layer – Это 2-битовое поле кодируется точно так же, как поле уровня в аудиопотоках, соответствующих ИСО/МЭК 13818-3 или ИСО/МЭК 11172-3 (см. § 2.4.2.3 стандарта ИСО/МЭК 13818-3). Уровень, указанный в этом поле, должен быть больше или равен наибольшему уровню, указанному в любом аудиокадре этого аудиопотока.

variable_rate_audio_indicator – Этот 1-битовый флаг, когда он установлен в '0' указывает, что в последующих аудиокадрах, которые должны воспроизводиться без нарушения непрерывности, закодированное значение поля двоичной скорости не должно изменяться.

2.6.6 Дескриптор иерархии

Дескриптор иерархии содержит информацию для идентификации программных элементов, содержащих компоненты иерархически кодированных видео и аудио потоков, а также конфиденциальных потоков (см. таблицу 2-49).

Таблица 2-49 – Дескриптор иерархии

Синтаксис	Количество битов	Мнемоника
hierarchy_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
зарезервирован	**4**	**bslbf**
hierarchy_type	**4**	**uimsbf**
зарезервирован	**2**	**bslbf**
hierarchy_layer_index	**6**	**uimsbf**
зарезервирован	**2**	**bslbf**
hierarchy_embedded_layer_index	**6**	**uimsbf**
зарезервирован	**2**	**bslbf**
hierarchy_channel	**6**	**uimsbf**
}		

2.6.7 Семантические определения полей в дескрипторе иерархии

hierarchy_type – Иерархическое соотношение между соответствующим уровнем иерархии и уровнем, иерархически вложенным в него, определяется в таблице 2-50.

hierarchy_layer_index – Индекс hierarchy_layer_index – это 6-битовое поле, которое определяет уникальный индекс соответствующего программного элемента в таблице иерархии уровня кодирования. В пределах определения отдельной программы индексы должны быть уникальными.

hierarchy_embedded_layer_index – Индекс hierarchy_embedded_layer_index – это 6-битовое поле, которое определяет индекс в иерархической таблице для программного элемента, который требуется оценить до декодирования элементарного потока, связанного с этим дескриптором hierarchy_descriptor. Это поле остается неопределенным, если значение hierarchy_type = 15 (базовый уровень).

hierarchy_channel – Поле hierarchy_channel – это 6-битовое поле, которое указывает номер канала в упорядоченном наборе каналов передачи, предназначенный для соответствующего программного элемента. Наиболее устойчивый канал передачи определяется наименьшим значением этого поля по отношению к общему определению иерархии передачи.

ПРИМЕЧАНИЕ. – Данный hierarchy_channel может быть одновременно назначен нескольким элементам программы.

Таблица 2-50 – Значения поля Hierarchy_type

Значение	Описание
0	Зарезервировано
1	Пространственная масштабируемость
2	Масштабируемость SNR
3	Временная масштабируемость
4	Секционирование данных
5	Расширение потока битов
6	Конфиденциальный поток
7	Многофункциональный профиль
8-14	Зарезервировано
15	Базовый уровень

2.6.8 Дескриптор регистрации

Дескриптор registration_descriptor предоставляет метод уникального и однозначного определения форматов конфиденциальной информации (см. таблицу 2-51).

Таблица 2-51 – Дескриптор регистрации

Синтаксис	Количество битов	Идентификатор
registration_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
format_identifier	**32**	**uimsbf**
for (i = 0; i < N; i++){		
additional_identification_info	**8**	**bslbf**
}		
}		

2.6.9 Семантические определения полей в дескрипторе регистрации

format_identifier – Идентификатор format_identifier – это 32-битовое значение, полученное от уполномоченного органа по регистрации, как обозначено в ИСО/МЭК JTC 1/SC 29.

additional_identification_info – Байты поля additional_identification_info, если таковые имеются, определяются владельцем этого идентификатора формата format_identifier, и не должны меняться после того, как они определены.

2.6.10 Дескриптор выравнивания информационного потока

Дескриптор выравнивания информационного потока описывает, какой тип выравнивания используется в связанном с ним элементарном потоке. Если индикатор data_alignment_indicator в заголовке пакета PES установлен в '1', и дескриптор присутствует, то требуется выравнивание, которое определено этим дескриптором (см. таблицу 2-52).

Таблица 2-52 – Дескриптор выравнивания информационного потока

Синтаксис	Количество битов	Мнемоника
data_stream_alignment_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
alignment_type	**8**	**uimsbf**
}		

2.6.11 Семантика полей в дескрипторе выравнивания информационного потока

alignment_type – В таблице 2-53 описываются типы выравнивания данных видеопотоков, соответствующих ИСО/МЭК 11172-2, Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, или видеопотоков, соответствующих ИСО/МЭК 14496-2 для случая, когда поле data_alignment_indicator в заголовке пакета PES имеет значение '1'. Для таких видеопотоков первый байт данных пакета PES (PES_packet_data_byte), следующий за заголовком PES, должен быть первым байтом стартового кода типа, показанного в таблице 2-53. В начале видеопоследовательности должно быть выполнено выравнивание стартового кода первого заголовка последовательности.

ПРИМЕЧАНИЕ. – Указание типа выравнивания '01' из таблицы 2-47 не исключает выравнивания с начала заголовка GOP или SEQ.

Определение модуля доступа приведено в § 2.1.1.

Таблица 2-53 – Значения выравнивания видеопотока

Тип выравнивания	Описание
00	Зарезервированы
01	Секция или модуль доступа видеопотока
02	Модуль доступа видеопотока
03	GOP или SEQ
04	SEQ
05-FF	Зарезервированы

В таблице 2-54 описывается тип выравнивания данных видеопотоков, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, когда поле data_alignment_indicator в заголовке пакета PES имеет значение '1'. В этом случае первый байт (PES_packet_data_byte), следующий после заголовка PES, должен быть первым байтом модуля доступа AVC или первым байтом секции AVC, как указано в значении поля alignment_type.

Таблица 2-54 – Значения выравнивания данных видеопотоков AVC

Тип выравнивания	Описание
00	Зарезервировано
01	Секция или модуль доступа AVC
02	Модуль доступа AVC
03-FF	Зарезервировано

В таблице 2-55 описывается тип выравнивания данных аудиопотоков, когда поле data_alignment_indicator в заголовке пакета PES имеет значение '1'. В таком случае первый байт (PES_packet_data_byte), следующий после заголовка PES, должен быть первым байтом синхрослова аудиопотока.

Таблица 2-55 – Значения выравнивания данных аудиопотоков

Тип выравнивания	Описание
00	Зарезервированы
01	Синхрослово
02-FF	Зарезервированы

2.6.12 Дескриптор целевой фоновой сетки

Возможен случай, когда один или несколько видеопотоков после декодирования не должны занимать всю область воспроизведения (например, монитора). Комбинация дескрипторов полей video_window_descriptor и target_background_grid_descriptor позволяет помещать эти окна видеоизображений в тех местах экрана, где их требуется воспроизвести. Дескриптор target_background_grid_descriptor используется для описания сетки в единицах пикселов, отображаемых на области дисплея. Дескриптор video_window_descriptor используется затем для описания (для соответствующего потока) точки на сетке, в которой должен располагаться верхний левый пиксел отображаемого окна или отображаемого прямоугольника видеомодуля представления, который должен быть воспроизведен. Это показано на рисунке 2-3.

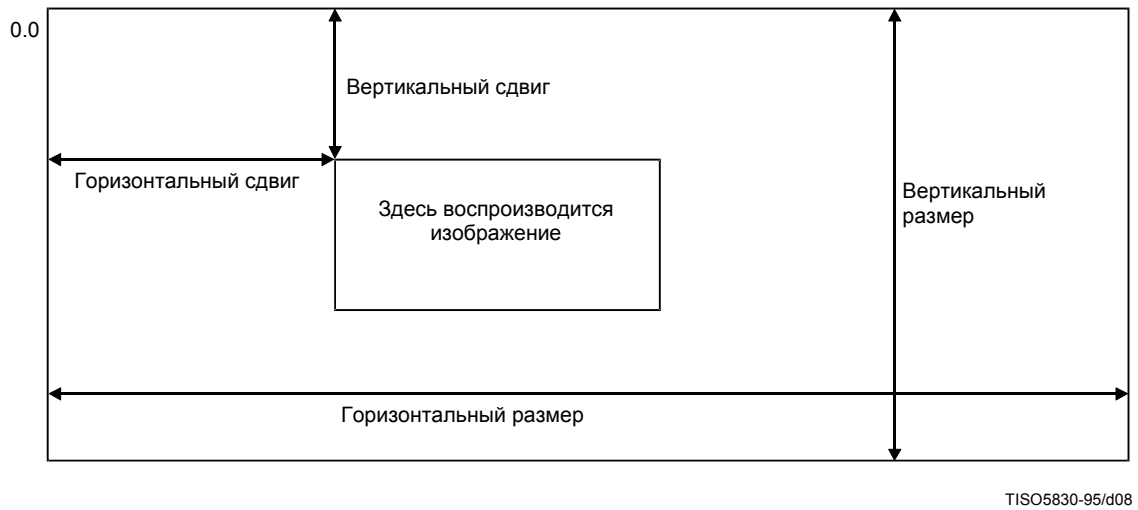


Рисунок 2-3 – Область воспроизведения в соответствии с дескриптором целевой фоновой сетки

2.6.13 Семантика полей в дескрипторе целевой фоновой сетки

horizontal_size – Горизонтальный размер целевой фоновой сетки в пикселах.

vertical_size – Вертикальный размер целевой фоновой сетки в пикселах.

aspect_ratio_information – Определяет соотношение геометрических размеров отсчета или соотношение геометрических размеров дисплея целевой фоновой сетки. Поле aspect_ratio_information определяется в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 (см. таблицу 2-56).

Таблица 2-56 – Дескриптор целевой фоновой сетки

Синтаксис	Количество битов	Мнемоника
target_background_grid_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
horizontal_size	**14**	**uimsbf**
vertical_size	**14**	**uimsbf**
aspect_ratio_information	**4**	**uimsbf**
}		

2.6.14 Дескриптор окна изображения

Дескриптор окна изображения используется для описания характеристик окна связанного с ним элементарного видеопотока. Его значения ссылаются на дескриптор целевой фоновой сетки этого же потока. Дескриптор target_background_grid_descriptor описан также в § 2.6.12 (см. таблицу 2-57).

Таблица 2-57 – Дескриптор окна изображения

Синтаксис	Количество битов	Мнемоника
video_window_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
horizontal_offset	**14**	**uimsbf**
vertical_offset	**14**	**uimsbf**
window_priority	**4**	**uimsbf**
}		

2.6.15 Семантические определения полей в дескрипторе окна изображения

horizontal_offset – Это значение указывает горизонтальное положение верхнего левого пиксела текущего воспроизводимого окна или воспроизводимого прямоугольника, если оно указано в расширении воспроизведения кадра на целевой фоновой сетке для воспроизведения, как определено в дескрипторе target_background_grid_descriptor. Верхний левый пиксел окна изображения должен быть одним из пикселов целевой фоновой сетки (см. рисунок 2-3).

vertical_offset – Это значение указывает вертикальное положение верхнего левого пиксела текущего воспроизводимого окна или воспроизводимого прямоугольника, если оно указано в расширении воспроизведения кадра на целевой фоновой сетке для воспроизведения, как определено в дескрипторе target_background_grid_descriptor. Верхний левый пиксел окна изображения должен быть одним из пикселов целевой фоновой сетки (см. рисунок 2-3).

window_priority – Это значение указывает, как перекрываются окна. Значение 0 имеет самый низший приоритет. А значение 15 имеет наивысший приоритет, т. е. окна с приоритетом 15 всегда видны.

2.6.16 Дескриптор условного доступа

Дескриптор условного доступа используется для определения как информации управления условным доступом на уровне всей системы, например сообщений, предоставляющих право доступа (EMM), и информации, определяемой элементарным потоком, например сообщений, управляющих правами доступа (ECM). Он может использоваться как в разделе TS_program_map_section (см. § 2.4.4.8), так и в карте program_stream_map (см. § 2.5.3). Если какой-либо элементарный поток скремблирован, то дескриптор условного доступа должен присутствовать для программы, содержащей этот элементарный поток. Если в Транспортном потоке присутствует какая-либо информация управления условным доступом на уровне всей системы, то дескриптор условного доступа должен присутствовать в таблице условного доступа.

Когда дескриптор условного доступа (CA) присутствует в карте TS_program_map_section (table_id = 0x02), идентификатор CA_PID указывает на пакеты, содержащие информацию управления доступом, относящуюся к программе, например, сообщения ECM. Его присутствие в виде информации программы указывает на его применимость к программе в целом. В этом же случае его присутствие в виде расширенной информации элементарного потока (ES) указывает на его применимость к соответствующему программному элементу. Предусмотрено также его применение и для конфиденциальной информации.

Когда дескриптор условного доступа (CA) присутствует в разделе CA_section (table_id = 0x01), идентификатор CA_PID указывает на пакеты, содержащие информацию управления на уровне всей системы и/или информацию управления доступом, например, сообщения EMM.

Содержание пакетов Транспортного потока, в которых содержится информация условного доступа, определяется в конфиденциальном порядке (см. таблицу 2-58).

Таблица 2-58 – Дескриптор условного доступа

Синтаксис	Количество битов	Мнемоника
CA_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
CA_system_ID	**16**	**uimsbf**
Зарезервировано	**3**	**bslbf**
CA_PID	**13**	**uimsbf**
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**uimsbf**
}		
}		

2.6.17 Семантические определения полей в дескрипторе условного доступа

CA_system_ID – Это 16-битовое поле, указывающее тип системы условного доступа (CA), применимой для соответствующих потоков ECM и/или EMM. Кодирование этого поля определяется в конфиденциальном порядке и не определяется в спецификациях МСЭ-Т | ИСО/МЭК.

CA_PID – Это 13-битовое поле, указывающее идентификатор PID пакетов Транспортного потока, которые должны содержать информацию либо ECM, либо EMM для системы условного доступа (CA), определенной соответствующим идентификатором CA_system_ID. Содержание пакетов (ECM или EMM), указанных идентификатором CA_PID, определяется из контекста, в котором находится CA_PID, т. е. в Транспортном потоке – разделом TS_program_map_section или таблицей CA, в Программном потоке – полем stream_id.

В Транспортных потоках наличие идентификатора PID 0x03 означает, что существует IPMP, описанный в стандарте ИСО/МЭК 13818-11, который используется компонентами этого Транспортного потока. В Программных потоках наличие идентификатора PID 0x00 означает, что IPMP, описанный в стандарте ИСО/МЭК 13818-11, используется компонентами этого Программного потока. В потоке, соответствующем настоящей Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, компоненты могут использовать как IPMP, описанный в стандарте ИСО/МЭК 13818-11, так и CA, определенный в стандарте ИСО/МЭК 13818-1:2000. Совместимость этих двух схем описана в стандарте ИСО/МЭК 13818-11.

2.6.18 Дескриптор языка, соответствующий ИСО 639

Дескриптор языка используется для определения языка связанного программного элемента (см. таблицу 2-59).

Таблица 2-59 – ИСО 639 Дескриптор языка

Синтаксис	Количество битов	Мнемоника
ISO_639_language_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
ISO_639_language_code	**24**	**bslbf**
audio_type	**8**	**bslbf**
}		
}		

2.6.19 Семантические определения полей в дескрипторе языка ИСО 639

ISO_639_language_code – Определяет язык или языки, используемые связанным с ним программным элементом. Поле ISO_639_language_code содержит 3-символьный код, определенный в стандарте ИСО 639, Часть 2. Каждый символ кодируется при помощи 8 битов в соответствии с ИСО 8859-1 и вводится в это 24-битовое поле в порядке очередности. В том случае, когда передаются аудиопотоки на нескольких языках, последовательность полей ISO_639_language_code должна отражать содержимое аудиопотока.

audio_type – Поле audio_type – это 8-битовое поле, которое указывает тип потока, определенный в таблице 2-60.

Таблица 2-60 – Значения типов аудиопотока

Значение	Описание
0x00	Неопределенный
0x01	Нет эффектов
0x02	Дефекты слуха
0x03	Комментарий для людей с дефектами зрения
0x04-0x7F	Конфиденциальный, определяется пользователем
0x80-0xFF	Зарезервированы

clean_effects – Это поле указывает, что рассматриваемый программный элемент не имеет языка.

hearing_impaired – Это поле указывает, что рассматриваемый программный элемент подготовлен для людей с дефектами слуха.

visual_impaired_commentary – Это поле указывает, что рассматриваемый программный элемент подготовлен для людей с дефектами зрения.

2.6.20 Дескриптор системных часов

Этот дескриптор содержит информацию о системных часах, которые использованы для формирования меток времени.

Если использовался внешний опорный генератор, то индикатор external_clock_reference_indicator может быть установлен в '1'. Декодер, дополнительно, может использовать внешний опорный генератор, если он доступен.

Если точность системных часов выше, чем требуемая точность 30×10^{-6}, то информация о точности часов может быть передана путем записи ее кода в полях clock_accuracy. Точность частоты системных часов равна:

$$clock_accuracy_integer \times 10^{-clock_accuracy_exponent} \times 10^{-6} \qquad (2\text{-}26)$$

Если значение clock_accuracy_integer установлено в '0', то точность системных часов равна 30×10^{-6}. Когда индикатор external_clock_reference_indicator установлен в '1', то точность системных часов относится к внешнему опорному генератору (см. таблицу 2-61).

Таблица 2-61 – Дескриптор системных часов

Синтаксис	Количество битов	Мнемоника
system_clock_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
external_clock_reference_indicator	**1**	**bslbf**
Зарезервировано	**1**	**bslbf**
clock_accuracy_integer	**6**	**uimsbf**
clock_accuracy_exponent	**3**	**uimsbf**
Зарезервировано	**5**	**bslbf**
}		

2.6.21 Семантические определения полей в дескрипторе системных часов

external_clock_reference_indicator – Это 1-битовый индикатор. Когда он установлен в '1', он указывает, что частота системные часы определяется частотой внешнего опорного генератора, который может быть доступен на декодере.

clock_accuracy_integer – Это 6-битовое целое число. Вместе со значением поля clock_accuracy_exponent оно определяет относительную погрешность частоты системных часов в долях от миллиона.

clock_accuracy_exponent – Это 3-битовое целое число. Вместе со значением поля clock_accuracy_integer оно определяет относительную погрешность частоты системных часов в долях от миллиона.

2.6.22 Дескриптор степени использования буфера мультиплексирования

Дескриптор степени использования буфера мультиплексирования определяет границы занятости буфера мультиплексирования декодера STD. Эта информация предназначена для таких устройств как ремультиплексоры, которые могут использовать эту информацию дл поддержания требуемой стратегии ремультиплексирования (см. таблицу 2-62).

Таблица 2-62 – Дескриптор степени использования буфера мультиплексирования

Синтаксис	Количество битов	Мнемоника
Multiplex_buffer_utilization_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
bound_valid_flag	**1**	**bslbf**
LTW_offset_lower_bound	**15**	**uimsbf**
Зарезервировано	**1**	**bslbf**
LTW_offset_upper_bound	**15**	**uimsbf**
}		

2.6.23 Семантические определения полей в дескрипторе степени использования буфера мультиплексирования

bound_valid_flag – Значение '1' указывает, что поля LTW_offset_lower_bound и LTW_offset_upper_bound действуют.

LTW_offset_lower_bound – Это 15-битовое поле определено только, если флаг bound_valid имеет значение '1'. Когда это поле определено, оно содержит единицы периодов часов (27 МГц/300), как определено для поля LTW_offset (см. § 2.4.3.4). Поле LTW_offset_lower_bound указывает наименьшее значение, которое могло бы иметь какое-либо поле LTW_offset, если бы это поле было закодировано в каждом пакете потока или потоков, указанных этим дескриптором. Когда дескриптор степени использования буфера мультиплексирования присутствует, реальные поля LTW_offset могут быть и могут не быть кодированы в виде двоичного потока. Эта граница действительна до следующего появления этого дескриптора.

LTW_offset_upper_bound – Это 15-битовое поле определено только, если флаг bound_valid имеет значение '1'. Когда это поле определено, оно содержит единицы периодов часов (27 МГц/300), как определено для поля LTW_offset (см. § 2.4.3.4). Поле LTW_offset_upper_bound указывает наибольшее значение, которое могло бы иметь какое-либо поле LTW_offset, если бы это поле было закодировано в каждом пакете потока или потоков,

указанных этим дескриптором. Когда дескриптор степени использования буфера мультиплексирования присутствует, реальные поля LTW_offset могут быть и могут не быть кодированы в виде двоичного потока. Эта граница действительна до следующего появления этого дескриптора.

2.6.24 Дескриптор авторских прав

Дескриптор copyright_descriptor предоставляет метод, позволяющий осуществлять идентификацию аудиовизуального произведения. Этот дескриптор copyright_descriptor относится к программам или программным элементам внутри программ (см. таблицу 2-63).

Таблица 2-63 – Дескриптор авторских прав

Синтаксис	Количество битов	Идентификатор
copyright_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
copyright_identifier	**32**	**uimsbf**
for (i = 0; i < N; i++){		
additional_copyright_info	**8**	**bslbf**
}		
}		

2.6.25 Семантические определения полей в дескрипторе авторских прав

copyright_identifier – Это поле представляет собой 32-битовое значение, полученное от уполномоченного органа по регистрации.

additional_copyright_info – Значения байтов поля additional_copyright_info, если таковые имеются, определяются владельцем этого идентификатора авторских прав copyright_identifier, и после того, как они определены, они не должны меняться.

2.6.26 Дескриптор максимальной двоичной скорости

См. таблицу 2-64.

Таблица 2-64 – Дескриптор максимальной двоичной скорости

Синтаксис	Количество битов	Идентификатор
maximum_bitrate_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
Зарезервировано	**2**	**bslbf**
maximum_bitrate	**22**	**uimsbf**
}		

2.6.27 Семантические определения полей в дескрипторе максимальной двоичной скорости

maximum_bitrate – Максимальная двоичная скорость кодируется в этом поле в виде 22-битового положительного целого числа. Это значение обозначает верхнюю границу двоичной скорости, включая транспортный заголовок, который будет введен в этот программный элемент или программу. Значение maximum_bitrate выражается в единицах по 50 байтов в секунду. Дескриптор maximum_bitrate_descriptor включен в Таблицу преобразования программы (PMT). Его наличие в виде расширенной программной информации указывает его применимость к программе в целом. Его наличие в виде информации элементарного потока (ES) указывает его применимость к связанному с ним программному элементу.

2.6.28 Дескриптор индикатора конфиденциальной информации

См. таблицу 2-65.

Таблица 2-65 – Дескриптор индикатора конфиденциальной информации

Синтаксис	Количество битов	Идентификатор
private_data_indicator_descriptor() {		
descriptor_tag	**38**	**uimsbf**
descriptor_length	**38**	**uimsbf**
private_data_indicator	**32**	**uimsbf**
}		

2.6.29 Семантические определения полей в дескрипторе индикатора конфиденциальной информации

private_data_indicator – Значение индикатора private_data_indicator конфиденциально и не должно определяться организациями МСЭ-Т | ИСО/МЭК.

2.6.30 Дескриптор буфера сглаживания

Этот дескриптор является дополнительным и содержит информацию о размере буфера сглаживания (SB_n), связанного с этим дескриптором, и о соответствующей скорости утечки из этого буфера для программного(ых) элемента(ов), к которому(ым) он относится.

В случае Транспортных потоков байты пакетов Транспортного потока соответствующего(их) программного(ых) элемента(ов), присутсвующего(их) в Транспортном потоке, являются входными для буфера SB_n с размером, определенным в поле sb_size, на момент времени, определенный уравнением 2-4.

В случае Программных потоков байты всех пакетов PES соответствующих элементарных потоков являются входными для буфера SB_n с размером, определенным в поле sb_size, на момент времени, определенный уравнением 2-21.

Когда в этом буфере присутствуют данные, байты удаляются из этого буфера со скоростью, определенной в поле sb_leak_rate. Буфер SB_n никогда не должен переполняться. Во время непрерывного существования программы, значение элементов дескриптора буфера сглаживания для различных программных элементов программы не должны изменяться.

Значение дескриптора буфера сглаживания buffer_descriptor определяется только, когда он включен в PMT или в карту Программного потока.

Если, в случае Транспортного потока, он присутствует в информации об элементарном потоке в Таблице преобразования программы, все пакеты Транспортного потока, имеющие идентификатор PID этого программного элемента, поступают в буфер сглаживания.

Если, в случае Транспортного потока, он присутствует в программной информации, то в буфер сглаживания поступают следующие пакеты Транспортного потока:

- все пакеты Транспортного потока со всеми значениями PID, перечисленными в виде полей elementary_PID в расширенной программной информации, а также;
- все пакеты Транспортного потока со значением PID = PMT_PID данной секции;
- все пакеты Транспортного потока со значением PCR_PID программы.

Все байты, которые поступают в соответствующий буфер, обязательно покидают его.

В любой заданный момент времени должно существовать не более одного дескриптора, указывающего на некоторый программный элемент, и не более одного дескриптора, указывающего на программу в целом.

Таблица 2-66 – Дескриптор буфера сглаживания

Синтаксис	Количество битов	Мнемоника
smoothing_buffer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
зарезервировано	**2**	**bslbf**
sb_leak_rate	**22**	**uimsbf**
зарезервировано	**2**	**bslbf**
sb_size	**22**	**uimsbf**
}		

2.6.31 Семантические определения полей в дескрипторе

sb_leak_rate – Это 22-битовое поле кодируется в виде положительного целого числа. Его содержание указывает значение скорости утечки из буфера SB_n для связанного с ним элементарного потока или других данных в единицах по 400 бит/с.

sb_size – Это 22-битовое поле кодируется в виде положительного целого числа. Его содержание указывает размер буфера сглаживания SB_n буфера мультиплексирование для связанного с ним элементарного потока или других данных в единицах по 1 байту (см. таблицу 2-66).

2.6.32 Дескриптор декодера STD

Этот дескриптор не является обязательным, он применяется только в модели декодера T-STD и к элементарным видеопотокам, соответствующим Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, и используется, как показано в § 2.4.2. Данный дескриптор к Программным потокам не применяется (см. таблицу 2-67).

Таблица 2-67 – Дескриптор декодера STD

Синтаксис	Количество битов	Мнемоника
STD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
зарезервировано	**7**	**bslbf**
leak_valid_flag	**1**	**bslbf**
}		

2.6.33 Семантические определения полей в дескрипторе декодера STD

leak_valid_flag – Флаг leak_valid_flag это 1-битовый флаг. Когда он установлен в '1', в декодере T-STD для передачи данных из буфера MB_n в буфер EB_n используется метод утечки, определенный в § 2.4.2.3. Если этот флаг имеет значение, равное '0', и поля vbv_delay, присутствующие в связанном с ним видеопотоке, не имеют значения 0xFFFF, для передачи данных из буфера MB_n в буфер EB_n используется метод vbv_delay, определенный в § 2.4.2.3.

2.6.34 Дескриптор IBP

Данный дополнительный дескриптор содержит информацию о некоторых характеристиках типов последовательности кадров в видеопотоке, соответствующем ИСО/МЭК 11172-2, Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 14496-2 (см. таблицу 2-68).

Таблица 2-68 – Дескриптор IBP

Синтаксис	Количество битов	Мнемоника
ibp_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
closed_gop_flag	**1**	**uimsbf**
identical_gop_flag	**1**	**uimsbf**
max_gop-length	**14**	**uimsbf**
}		

2.6.35 Семантические определения полей в дескрипторе IBP

closed_gop_flag – Этот 1-битовый флаг, когда он установлен в '1', указывает, что заголовок группы кадров кодируется перед каждым кадром I, и что флаг closed_gop установлен в ‘1' во всех заголовках групп кадров в видеопоследовательности.

identical_gop_flag – Этот 1-битовый флаг, когда он установлен в '1' указывает, что количество кадров P и кадров B между кадрами I, а также типы кодирования изображения, и типы последовательности изображений между кадрами I остаются одинаковыми на протяжении всей последовательности, за исключением, возможно, изображений до второго кадра I.

max_gop_length – Это 14-битовое целое число без знака указывает максимальное количество кодированных кадров изображения между любыми двумя последовательными кадрами I в последовательности. Значение '0' является запрещенным.

2.6.36 Дескриптор видеопотока MPEG-4

Для отдельных потоков, соответствующих ИСО/МЭК 14496-2, передаваемых непосредственно в пакетах PES, как определено в § 2.11.2, дескриптор видеопотока MPEG-4 содержит базовую информацию для идентификации параметров кодирования этих элементарных видеопотоков. Дескриптор видеопотока MPEG-4 не применяется к потокам ИСО/МЭК 14496-2, инкапсулированным в пакеты SL и пакеты FlexMux, как определено в § 2.11.3.

Таблица 2-69 – Дескриптор видеопотока MPEG-4

Синтаксис	Количество битов	Мнемоника
MPEG-4_video_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MPEG-4_visual_profile_and_level	**8**	**uimsbf**
}		

2.6.37 Семантические определения полей в дескрипторе видеопотока MPEG-4

MPEG-4_video_profile_and_level – Это 8-битовое поле должно определять профиль и уровень видеопотока, соответствующего ИСО/МЭК 14496-2. Это поле должно кодироваться тем же значением, что и поле profile_and_level_indication в заголовке последовательности визуальных объектов в связанном с ним потоке, соответствующем ИСО/МЭК 14496-2.

2.6.38 Дескриптор аудиопотока MPEG-4

Для отдельных потоков, соответствующих ИСО/МЭК 14496-3, передаваемых непосредственно в пакетах PES, как определено в § 2.11.2, дескриптор аудиопотока MPEG-4 содержит базовую информацию для идентификации параметров кодирования этих элементарных аудиопотоков. Дескриптор аудиопотока MPEG-4 не применяется к потокам ИСО/МЭК 14496-3, инкапсулированным в пакеты SL и пакеты FlexMux, как определено в § 2.11.3.

Таблица 2-70 – Дескриптор аудиопотока MPEG-4

Синтаксис	Количество битов	Мнемоника
MPEG-4_audio_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MPEG-4_audio_profile_and_level	**8**	**uimsbf**
}		

2.6.39 Семантические определения полей в дескрипторе аудиопотока MPEG-4

MPEG-4_audio_profile_and_level – Это 8-битовое поле должно определять профиль и уровень аудиопотока, соответствующего ИСО/МЭК 14496-3, в соответствии с таблицей 2-71.

Таблица 2-71 – Значения MPEG-4_audio_profile_and_level

Значение	Описание
0x00-0x0F	Зарезервированы
0x10	Основной профиль, уровень 1
0x11	Основной профиль, уровень 2
0x12	Основной профиль, уровень 3
0x13	Основной профиль, уровень 4
0x14-0x17	Зарезервированы
0x18	Масштабируемый профиль, уровень 1
0x19	Масштабируемый профиль, уровень 2
0x1A	Масштабируемый профиль, уровень 3
0x1B	Масштабируемый профиль, уровень 4
0x1C-0x1F	Зарезервированы

Таблица 2-71 – Значения MPEG-4_audio_profile_and_level

Значение	Описание
0x20	Речевой профиль, уровень 1
0x21	Речевой профиль, уровень 2
0x22-0x27	Зарезервированы
0x28	Синтезированный профиль, уровень 1
0x29	Синтезированный профиль, уровень 2
0x2A	Синтезированный профиль, уровень 3
0x2B-0x2F	Зарезервированы
0x30	Высококачественный аудиопрофиль, уровень 1
0x31	Высококачественный аудиопрофиль, уровень 2
0x32	Высококачественный аудиопрофиль, уровень 3
0x33	Высококачественный аудиопрофиль, уровень 4
0x34	Высококачественный аудиопрофиль, уровень 5
0x35	Высококачественный аудиопрофиль, уровень 6
0x36	Высококачественный аудиопрофиль, уровень 7
0x37	Высококачественный аудиопрофиль, уровень 8
0x38	Аудиопрофиль с малой задержкой, уровень 1
0x39	Аудиопрофиль с малой задержкой, уровень 2
0x3A	Аудиопрофиль с малой задержкой, уровень 3
0x3B	Аудиопрофиль с малой задержкой, уровень 4
0x3C	Аудиопрофиль с малой задержкой, уровень 5
0x3D	Аудиопрофиль с малой задержкой, уровень 6
0x3E	Аудиопрофиль с малой задержкой, уровень 7
0x3F	Аудиопрофиль с малой задержкой, уровень 8
0x40	Аудиопрофиль естественного звучания, уровень 1
0x41	Аудиопрофиль естественного звучания, уровень 2
0x42	Аудиопрофиль естественного звучания, уровень 3
0x43	Аудиопрофиль естественного звучания, уровень 4
0x44-0x47	Зарезервированы
0x48	Профиль взаимодействия с подвижной радиосвязью, уровень 1
0x49	Профиль взаимодействия с подвижной радиосвязью, уровень 2
0x4A	Профиль взаимодействия с подвижной радиосвязью, уровень 3
0x4B	Профиль взаимодействия с подвижной радиосвязью, уровень 4
0x4C	Профиль взаимодействия с подвижной радиосвязью, уровень 5
0x4D	Профиль взаимодействия с подвижной радиосвязью, уровень 6
0x4E-0x4F	Зарезервированы
0x50	Профиль AAC, уровень 1
0x51	Профиль AAC, уровень 2
0x52	Профиль AAC, уровень 4
0x53	Профиль AAC, уровень 5
0x54-0x57	Зарезервированы
0x58	Профиль AAC с высокой эффективностью, уровень 2
0x59	Профиль AAC с высокой эффективностью, уровень 3
0x5A	Профиль AAC с высокой эффективностью, уровень 4
0x5B	Профиль AAC с высокой эффективностью, уровень 5
0x5C-0xFF	Зарезервированы

2.6.40 Дескриптор IOD

Дескриптор IOD содержит структуру "исходный дескриптор объекта" (InitialObjectDescriptor). Исходный дескриптор объекта позволяет получить доступ к множеству потоков стандарта ИСО/МЭК 14496 путем идентификации значений ES_ID потоков описания сцены и дескрипторов объекта, соответствующих ИСО/МЭК 14496-1. Как поток описания сцены, так и поток дескрипторов содержат дополнительную информацию о потоках, соответствующих ИСО/МЭК 14496, которые являются составной частью этой сцены. Процедура доступа к содержанию описана в Приложении R. Дескриптор InitialObjectDescriptor определен в § 8.6.3 стандарта ИСО/МЭК 14496-1.

В Транспортном потоке дескриптор IOD должен передаваться по цепочке дескриптора, которая в Таблице преобразования программы следует сразу после поля program_info_length. Если в Программном потоке присутствует карта Программного потока, то дескриптор IOD должен передаваться по цепочке дескриптора, которая в Карте Программного потока следует сразу после поля program_stream_info_length. С программой может быть связано несколько дескрипторов IOD.

ПРИМЕЧАНИЕ. – В настоящей Спецификации не определяется то, как метки IOD_label могут использоваться информационными службами более высокого уровня для однозначного выбора одного из вариантов воспроизведения, соответствующих ИСО/МЭК 14496, определенных несколькими дескрипторами IOD.

Таблица 2-72 – IOD descriptor

Синтаксис	Количество битов	Мнемоника
IOD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
Scope_of_IOD_label	**8**	**uimsbf**
IOD_label	**8**	**uimsbf**
InitialObjectDescriptor ()	**8**	**uimsbf**
}		

2.6.41 Семантические определения полей в дескрипторе IOD

Scope_of_IOD_label – Это 8-битовое поле определяет сферу применения поля IOD_label. Значение 0x10 указывает, что метка IOD_label является уникальной в рамках Программного потока или в рамках определенной программы в Транспортном потоке, в котором передается дескриптор IOD. Значение 0x11 указывает, что метка IOD_label является уникальной в рамках Транспортного потока, в котором передается дескриптор IOD. Все другие значения поля scope_of_IOD_label зарезервированы.

IOD_label – Это 8-битовое поле определяет метку дескриптора IOD.

InitialObjectDescriptor () – Эта структура определена в § 8.6.3.1 стандарта ИСО/МЭК 14496-1.

2.6.42 Дескриптор SL

Дескриптор SL должен использоваться, когда отдельный поток, соответствующий ИСО/МЭК 14496-1, и сформированный в виде пакетов SL, инкапсулирован в пакеты PES. В случае Транспортного потока дескриптор SL связывает идентификатор ES_ID этого потока, сформированного в виде пакетов SL, с идентификатором elementary_PID, а в случае Программного потока – с идентификатором elementary_stream_id. В Транспортном потоке дескриптор SL должен передаваться для соответствующего элементарного потока в цепочке дескриптора, которая следует в Таблице преобразования программы сразу после поля ES_info_length. Если в Программном потоке присутствует карта Программного потока, то дескриптор SL должен передаваться в цепочке дескриптора, которая следует в карте Программного потока сразу после поля elementary_stream_info_length.

ПРИМЕЧАНИЕ. – Потоки, сформированные в виде пакетов SL, могут использоваться в Программном потоке. Однако, идентификатор stream_id существует только для потоков, сформированных в виде пакетов SL и соответствующих ИСО/МЭК 14496-1. Для того чтобы в Программном потоке связать несколько таких потоков со сценой, соответствующей ИСО/МЭК 14496-1, должен использоваться FlexMux, и об этом должно быть соответствующим образом сообщено в дескрипторе FMC. Такого ограничения не существует в Транспортном потоке, где дескриптор SL обеспечивает однозначное преобразование между значением ES_ID, соответствующим ИСО/МЭК 14496-1, и значением elementary_PID, соответствующим Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

Таблица 2-73 – Дескриптор SL

Синтаксис	Количество битов	Мнемоника
SL_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
ES_ID	**16**	**uimsbf**
}		

2.6.43 Семантические определения полей в дескрипторе SL

ES_ID – Это 16-битовое поле должно определять идентификатор потока, сформированного в виде пакетов SL и соответствующего ИСО/МЭК 14496-1.

2.6.44 Дескриптор FMC

Дескриптор FMC указывает, что до инкапсуляции в пакеты PES или секции ISO/IEC14496_sections для мультиплексирования в единый поток FlexMux потоков, сформированных в виде пакетов SL и соответствующих ИСО/МЭК 14496-1, использовался метод FlexMux, соответствующий ИСО/МЭК 14496-1. Дескриптор FMC в потоке FlexMux связывает каналы FlexMux со значениями идентификаторов ES_ID потоков, сформированных в виде пакетов SL.

В Транспортном потоке дескриптор FMC требуется для каждого программного элемента, указанного значением идентификатора elementary_PID, и в Программном потоке, в котором передается поток FlexMux, дескриптор FMC требуется для каждого идентификатора elementary_stream_id. В Транспортном потоке дескриптор FMC для соответствующего элементарного потока должен передаваться в цепочке дескриптора, которая следует в Таблице преобразования программы сразу после поля ES_info_length. Если в Программном потоке присутствует карта Программного потока, то дескриптор FMC должен передаваться в цепочке дескриптора, которая следует в карте Программного потока сразу после поля elementary_stream_info_length.

Для каждого потока, сформированного в виде пакетов SL, в потоке FlexMux при помощи отдельной записи в дескрипторе FMC должен быть идентифицирован канал FlexMux.

Таблица 2-74 – Дескриптор FMC

Синтаксис	Количество битов	Мнемоника
FMC_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < descriptor_length; i + = 3) {		
ES_ID	**16**	**uimsbf**
FlexMuxChannel	**8**	**uimsbf**
}		
}		

2.6.45 Семантические определения полей в дескрипторе FMC

ES_ID – Это 16-битовое поле определяет идентификатор потока, сформированного в виде пакетов SL и соответствующего ИСО/МЭК 14496-1.

FlexMuxChannel – Это 8-битовое поле определяет номер канала FlexMux, используемого для этого потока, сформированного в виде пакетов SL.

2.6.46 Дескриптор External_ES_ID

Дескриптор External_ES_ID присваивает идентификатор ES_ID, определенный в ИСО/МЭК 14496-1, программному элементу, которому другими средствами никакого значения ES_ID присвоено не было. Этот идентификатор ES_ID позволяет указывать компоненты описания сцены, не соответствующие ИСО/МЭК 14496, или, например, связывать компонент, не соответствующий ИСО/МЭК 14496, с потоком IPMP.

В Транспортном потоке присвоение идентификатора ES_ID должно осуществляться при помощи передачи дескриптора External_ES_ID соответствующего элементарного потока в цепочке дескриптора, которая следует в Таблице преобразования программы сразу после поля ES_info_length. Если в Программном потоке присутствует карта Программного потока, то дескриптор External_ES_ID должен передаваться в цепочке дескриптора, которая следует сразу после поля elementary_stream_info_length в карте Программного потока.

Таблица 2-75 – Дескриптор External_ES_ID

Синтаксис	Количество битов	Мнемоника
External_ES_ID_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
External_ES_ID	**16**	**uimsbf**
}		

2.6.47 Семантические определения полей в дескрипторе External_ES_ID

External_ES_ID – Это 16-битовое поле присваивает компоненту программы идентификатор ES_ID, определенный в ИСО/МЭК 14496-1.

2.6.48 Дескриптор Muxcode

Дескриптор Muxcode передает структуры записи MuxCodeTableEntry, определенные в § 11.2.4.3 стандарта ИСО/МЭК 14496-1. Записи MuxCodeTableEntries конфигурируют режим MuxCode потока FlexMux.

Каждому идентификатору elementary_PID или elementary_stream_id потока FlexMux, соответствующего ИСО/МЭК 14496-1, в котором используется режим MuxCode, может быть присвоен один или несколько дескрипторов Muxcode, соответственно. В Транспортном потоке дескриптор Muxcode для соответствующего элементарного потока должен передаваться в цепочке дескриптора, которая следует в Таблице преобразования программы сразу после поля ES_info_length. Если в Программном потоке присутствует карта Программного потока, то дескриптор Muxcode должен передаваться в цепочке дескриптора, которая следует в карте Программного потока сразу после поля elementary_stream_info_length.

Записи MuxCodeTableEntries могут заменяться новыми версиями. В случае такого обновления поле version_number каждой Таблицы преобразования программы или поле program_stream_map_version каждой карты Программного потока, соответственно, в которых – в их цепочке дескриптора – передается дескриптор Muxcode, должны быть увеличены на 1 (по модулю 32).

Таблица 2-76 – Дескриптор Muxcode

Синтаксис	Количество битов	Мнемоника
Muxcode_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
MuxCodeTableEntry ()		
}		
}		

2.6.49 Семантические определения полей в дескрипторе Muxcode

MuxCodeTableEntry () – Эта структура определена в § 11.2.4.3 стандарта ИСО/МЭК 14496-1.

2.6.50 Дескриптор FmxBufferSize

Дескриптор FmxBufferSize передает информацию о размере буфера FlexMux (FB) для каждого потока, сформированного в виде пакетов SL, который мультиплексирован в потоке FlexMux.

Каждому идентификатору elementary_PID или elementary_stream_id, передающему поток FlexMux, соответствующий ИСО/МЭК 14496-1 должен быть присвоен один дескриптор FmxBufferSize. В Транспортном потоке дескриптор FmxBufferSize для соответствующего элементарного потока должен передаваться в цепочке дескриптора, которая следует в Таблице преобразования программы сразу после поля ES_info_length. Если в Программном потоке присутствует карта Программного потока, то дескриптор FmxBufferSize должен передаваться в цепочке дескриптора, которая следует в карте Программного потока сразу после поля elementary_stream_info_length.

Таблица 2-77 – Дескриптор FmxBufferSize

Синтаксис	Количество битов	Мнемоника
FmxBufferSize_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
DefaultFlexMuxBufferDescriptor()		
for (i=0; i<descriptor_length; i += 4) {		
FlexMuxBufferDescriptor()		
}		
}		

2.6.51 Семантические определения полей в дескрипторе FmxBufferSize

FlexMuxBufferDescriptor() – Этот дескриптор определяет размер буфера FlexMux для одного потока, который сформирован в виде пакетов SL и передается в потоке FlexMux. Он определен в § 11.2 стандарта ИСО/МЭК 14496-1.

DefaultFlexMuxBufferDescriptor() – Этот дескриптор определяет размер буфера для этого потока FlexMux. Он определен в §11.2 стандарта ИСО/МЭК 14496-1.

2.6.52 Дескриптор MultiplexBuffer

Дескриптор MultiplexBuffer передает информацию о размере буфера MB_n, а также о скорости утечки Rx_n. с которой данные из транспортного буфера TB_n передаются в буфер MB_n для определенного программного элемента, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, и указанного в Таблице преобразования программы значением идентификатора elementary_PID.

Каждому идентификатору elementary_PID, который содержит поток FlexMux, соответствующий ИСО/МЭК 14496, или поток, сформированный в виде пакетов SL, включая пакеты, содержащие секции ISO_IEC_14496_sections, должен быть присвоен один дескриптор MultiplexBuffer. Определения буферов и скоростей в модели декодера T-STD для декодирования содержания, соответствующего стандарту ИСО/МЭК 14496, приведены в § 2.11.3.9.

Дескриптор MultiplexBuffer должен передаваться в цепочке дескриптора, которая следует в Таблице преобразования программы сразу после поля ES_info_length.

Таблица 2-78 – Дескриптор MultiplexBuffer

Синтаксис	Количество битов	Мнемоника
MultiplexBuffer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MB_buffer_size	**24**	**uimsbf**
TB_leak_rate	**24**	**uimsbf**
}		

2.6.53 Семантические определения полей в дескрипторе MultiplexBuffer

MB_buffer_size – Это 24-битовое поле должно определять (в байтах) размер буфера MB_n элементарного потока "n", который связан с этим дескриптором.

TB_leak_rate – Это 24-битовое поле должно определять (в единицах по 400 бит/с) скорость, с которой данные передаются из транспортного буфера TB_n в буфер мультиплексирования MB_n для элементарного потока "n", который связан с этим дескриптором.

2.6.54 Дескриптор FlexMuxTiming

См. таблицу 2-79.

Таблица 2-79 – Дескриптор FlexMuxTiming

Синтаксис	Количество битов	Мнемоника
FlexMuxTiming_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
FCR_ES_ID	**16**	**uimsbf**
FCRResolution	**32**	**uimsbf**
FCRLength	**8**	**uimsbf**
FmxRateLength	**8**	**uimsbf**
}		

2.6.55 Семантические определения полей в дескрипторе FlexMuxTiming

FCR_ES_ID – Это идентификатор ES_ID, связанный с этим потоком опорных тактовых импульсов.

FCRResolution – Это разрешение временной базы объекта в числе циклов в секунду.

FCRLength – Это продолжительность поля fmxClockReference в пакетах FlexMux с индексом = 238. Продолжительность поля = 0 должна означать, что в этом потоке FlexMux нет ни одного пакета FlexMux с индексом = 238. Поле FCRlength должно принимать значения между нулем и 64.

FmxRateLength – Это продолжительность поля fmxRate в пакетах FlexMux с индексом = 238. Поле FmxRateLength должно принимать значения между 1 и 32.

2.6.56 Дескриптор маркировки содержания

Дескриптор маркировки содержания присваивает метку содержанию; эта метка может использоваться метаданными для указания связанного с ними содержания. Эта метка (content_reference_id_record) имеет формат, определенный специально для использования метаданных. Дескриптор маркировки содержания связывается с сегментом содержания. В настоящем разделе сегмент содержания определяется следующим образом "часть времени программы, элементарного потока (аудио или видео) или любая комбинация программ или элементарных потоков". Этот дескриптор может быть включен в PMT в цепочке дескриптора либо для программы, либо для элементарного потока, но может также содержаться в таблицах, которые в настоящей Спецификации не определены, например таблицах для описания сегментов программ или элементарных потоков. Кроме того, дескриптор маркировки содержания содержит информацию о том, какая временная база содержания используется, и каков сдвиг между временной базой содержания и временной базой метаданных. Когда в качестве временной базы содержания используется концепция управления и контроля цифровой среды хранения данных (DSM-CC) с нормальным временем воспроизведения (Normal Play Time (NPT)), определенная в ИСО/МЭК 13818-6, в дескрипторе содержится ID временной базы NPT. Этот дескриптор позволяет передавать конфиденциальную информацию. См. таблицу 2-80.

Таблица 2-80 – Дескриптор маркировки содержания

Синтаксис	Количество битов	Мнемоника
Content_labeling_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format== 0xFFFF){		
metadata_application_format_identifier	**32**	**uimsbf**
}		
content_reference_id_record_flag	**1**	**bslbf**
content_time_base_indicator	**4**	**uimsbf**
зарезервировано	**3**	**bslbf**
if (content_reference_id_record_flag == ‘1'){		
content_reference_id_record_length	**8**	**uimsbf**
for (i=0; i<content_reference_id_record_length;i++){		
content_reference_id_byte	**8**	**bslbf**
}		
}		
if (content_time_base_indicator== 1|2){		
зарезервировано	**7**	**bslbf**
content_time_base_value	**33**	**uimsbf**
зарезервировано	**7**	**bslbf**
metadata_time_base_value	**33**	**uimsbf**
}		
if (content_time_base_indicator== 2){		
зарезервировано	**1**	**bslbf**
contentId	**7**	**uimsbf**
}		
if (content_time_base_indicator==3|4|5|6|7){		
time_base_association_data_length	**8**	**uimsbf**
for (i=0; i< time_base_association_data_length;i++){		
зарезервировано	**8**	**bslbf**
}		
}		
for (i=0; i<N;i++){		
private_data_byte	**8**	**bslbf**
}		
}		

2.6.57 Семантические определения полей в дескрипторе маркировки содержания

metadata_application_format: Поле metadata_application_format– это 16-битовое поле, закодированное в соответствии с определениями, показанными в таблице 2-81, которая определяет приложение, ответственное за определение использования, синтаксиса и семантики записи content_reference_id и любых других полей, определенных в этом дескрипторе в конфиденциальном порядке. См. также § 2.12.1. Значение 0xFFFF указывает, что данные о формате передаются при помощи значения, передаваемого в поле metadata_application_format_identifier.

Таблица 2-81 – Metadata_application_format

Значение	Описание
0x0000-0x000F	Зарезервированы
0x0010	ИСО 15706 (ISAN), закодированное в двоичной форме (см. Примечания 1 и 3)
0x0011	ИСО 15706-2 (V-ISAN), закодированное в двоичной форме (см. Примечания 2 и 3)
0x0012-0x00FF	Зарезервированы
0x0100-0xFFFE	Определяется пользователем
0xFFFF	Определяется полем metadata_application_format_identifier
ПРИМЕЧАНИЕ 1. – Для ISAN байт content_reference_id_byte установлен в двоичное кодирование, а значение content_reference_id_record_length установлено равным 0x08. ПРИМЕЧАНИЕ 2. – Для V-ISAN байт content_reference_id_byte установлен в двоичное кодирование, а значение content_reference_id_record_length установлено равным 0x0C. ПРИМЕЧАНИЕ 3. – Для взаимодействия между приложениями метаданных, в которых используются значения формата metadata_application_format 0x0010 и 0x0011, рекомендуется, чтобы флаг content_reference_id_flag был бы установлен в '1', а индикатор content_time_base_indicator был бы установлен в '00'.	

metadata_application_format_identifier: Кодирование этого 32-битового поля полностью эквивалентно кодированию поля format_identifier в дескрипторе регистрации registration_descriptor, которое определено в § 2.6.8.

ПРИМЕЧАНИЕ. – Уполномоченным органом регистрации полей format_identifier является Общество кино и телеинженеров (SMPTE).

content_reference_id_record_flag: Флаг content_reference_id_record_flag – это 1-битовый флаг, который сигнализирует о присутствии в этом дескрипторе записи content_reference_id_record.

content_time_base_indicator: Индикатор content_time_base_indicator – это 4-битовое поле, которое определяет используемую временную базу содержания. Если этот дескриптор связан с программой, то временная база содержания применяется ко всем потокам, которые являются частью этой программы. Значение '1' указывает использование STC, тогда как значение '2' указывает использование NPT, т. е. нормального времени воспроизведения, которое определено в ИСО/МЭК 13818-6. Значения между 8 и 15 указывают использование временной базы содержания, которая определена в конфиденциальном порядке. Если этот индикатор закодирован со значением '0', то в этом дескрипторе не определено никакой временной базы содержания. Если для программы или потока не определено никакой временной базы содержания, то преобразование тактовых импульсов метаданных в содержание в настоящей Спецификации также не определяется.

Таблица 2-82 – Значения индикатора content_time_base_indicator

Значение	Описание
0	В этом дескрипторе временная база содержания не определена
1	Использование STC
2	Использование NPT
3-7	Зарезервированы
8-15	Использование временной базы содержания, определенной в конфиденциальном порядке

content_reference_id_record_length: Поле content_reference_id_record_length – это 8-битовое поле, которое указывает количество байтов content_reference_id_byte, которые следуют сразу после этого поля. Это поле не должно быть закодировано со значением '0'.

content_reference_id_byte: Байт content_reference_id_byte – это участок строки из одного или нескольких последовательных байтов, которая присваивает содержанию, с которым связан этот дескриптор, один или несколько опорных идентификаторов. Формат этой строки байтов определяется символом, который указан значением, закодированным в поле metadata_application_format.

content_time_base_value: Поле content_time_base_value – это 33-битовое поле, которое определяет значение (в единицах по 90 кГц) временной базы содержания, указанной полем content_time_base_indicator.

metadata_time_base_value: Поле metadata_time_base_value – это 33-битовое поле, которое закодировано в единицах по 90 кГц. Это поле содержит закодированное значение временной базы метаданных в момент времени, когда временная база, указанная индикатором content_time_base_indicator, достигает значения, закодированного в поле content_time_base_value. Отметим, что временная база метаданных может использовать любой масштаб времени, но, что ее значение должно быть закодировано в единицах по 90 кГц. Например, если используется временной код типа SMPTE, тогда количество часов, минут, секунд и кадров выражается соответствующим количеством единиц по 90 кГц.

contentId: Поле contentId – это 7-битовое поле, которое указывает значение поля content_Id в Справочном дескрипторе NPT для применяемой временной базы NPT.

time_base_association_data_length: Поле time_base_association_data_length – это 8-битовое поле, которое определяет количество зарезервированных байтов, которые следуют сразу после этого поля. Зарезервированные байты могут использоваться для передачи данных, связанных с временной базой для тех временных баз, которые будут определены в будущем.

private_data_byte: Поле private_data_byte – это 8-битовое поле. Байты private_data_byte представляют собой данные, формат которых определяется конфиденциальным порядком. Эти байты могут использоваться для предоставления дополнительной информации, которая представляется необходимой. Использование этих байтов определяется форматом приложения метаданных.

2.6.58 Дескриптор указателя метаданных

Дескриптор указателя метаданных указывает отдельную услугу метаданных и связывает эту услугу метаданных с аудиовизуальным содержанием потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Метаданные связаны с содержанием в контексте дескриптора. Контекст определяется местоположением дескриптора. В Транспортном потоке дескриптор может располагаться в таблице PMT в цепочке дескриптора либо для программы, либо для элементарного потока, но может также располагаться в таблицах, которые в настоящей Спецификации не определены, например, в таблицах, описывающих услуги "из пункта во множество пунктов" или радиовещательные услуги. Метаданные могут располагаться в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, но те же самые метаданные могут быть представлены и в иных местоположениях, например, в интернете.

Дескриптор может содержать информацию о местоположении метаданных, которые не передаются в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1; кодирование информации о местоположении определяется форматом приложения метаданных. Дескриптор позволяет передавать конфиденциальную информацию.

Для метаданных, передаваемых в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, дескриптор определяет инструменты, используемые для такой передачи. Если метаданные передаются в пакетах PES, секциях метаданных или синхронизированных секциях загрузки, соответствующих ИСО/МЭК 13818-6, то поле metadata_service_id определяет услугу метаданных в указанном потоке метаданных. Если для передачи метаданных используется карусель, соответствующая ИСО/МЭК 13818-6, то конфиденциальная информация может предоставлять информацию для создания сообщения об услуге метаданных, например, о применяемом значении идентификатора module_id для передачи метаданных в составе данных карусельного типа, и об имени файла метаданных, когда используется объект карусельного типа.

Приемники должны быть осведомлены о том, что услуги метаданных могут быть указаны из одной и той же программы или аудиовизуального потока (определенными контекстом дескриптора). Для указания каждой услуги метаданных, используемой программой или аудиовизуальным потоком, должен использоваться уникальный дескриптор указателя метаданных. Аналогично, одна и та же услуга метаданных может быть указана из нескольких программ или аудиовизуальных потоков за счет использования для каждой связи отдельных дескрипторов указателей метаданных.

Таблица 2-83 – Дескриптор указателя метаданных

Синтаксис	Количество битов	Мнемоника
Metadata_pointer_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format== 0xFFFF){		
metadata_application_format_identifier	**32**	**uimsbf**
}		
metadata_format	**8**	**uimsbf**
if (metadata_format== 0xFF){		
metadata_format_identifier	**32**	**uimsbf**
}		
metadata_service_id	**8**	**uimsbf**
metadata_locator_record_flag	**1**	**bslbf**
MPEG_carriage_flags	**2**	**uimsbf**
зарезервировано	**5**	**bslbf**
if (metadata_locator_record_flag == '1'){		
metadata_locator_record_length	**8**	**uimsbf**
for (i = 0; i < metadata_locator_record_length; i + +){		
metadata_locator_record_byte	**8**	**bslbf**
}		
}		

Синтаксис	Количество битов	Мнемоника
if (MPEG_carriage_flags == 0|1|2){		
program_number	**16**	**uimsbf**
}		
if (MPEG_carriage_flags == 1){		
transport_stream_location	**16**	**uimsbf**
transport_stream_id	**16**	**uimsbf**
}		
for (i=0; i<N;i++){		
private_data_byte	**8**	**bslbf**
}		
}		

2.6.59 Семантические определения полей в дескрипторе указателя метаданных

metadata_application_format: Поле metadata_application_format – это 16-битовое поле, которое определяет приложение, ответственное за определение использования, синтаксиса и семантики записи metadata_locator_record и любых других полей в этом дескрипторе, которые определены в конфиденциальном порядке. Кодирование этого поля определяется в таблице 2-81 в § 2.6.57.

metadata_application_format_identifier: Кодирование этого поля определяется в § 2.6.57.

metadata_format: Поле metadata_format – это 8-битовое поле, которое указывает формат и кодирование метаданных. Кодирование этого поля определяется в таблице 2-84.

Таблица 2-84 – Значения формата метаданных

Значение	Описание
0x00-0x0F	Зарезервированы
0x10	ИСО/МЭК 15938-1 TeM
0x11	ИСО/МЭК 15938-1 BiM
0x12-0x3E	Зарезервированы
0x3F	Определяется форматом приложения метаданных
0x40-0xFE	Конфиденциальное использование
0xFF	Определяется полем metadata_format_identifier

Значения 0x10 и 0x11 указывают данные, определенные в ИСО/МЭК 15938-1. Значение 0x3F указывает, что формат определяется содержанием, указанным в поле metadata_application_format. Значения в диапазоне от 0x40 до 0xFE включительно доступны для использования с целью сообщения о форматах, определенных в конфиденциальном порядке. Значение 0xFF указывает, информация о формате передается в поле metadata_format_identifier.

metadata_format_identifier: Кодирование этого 32-битового поля полностью эквивалентно кодированию поля format_identifier в дескрипторе регистрации (registration_descriptor), которое определено в § 2.6.8.

ПРИМЕЧАНИЕ. – Уполномоченным органом регистрации поля format_identifier является Общество кино и телеинженеров (SMPTE).

metadata_service_id: Это 8-битовое поле указывает услугу метаданных. Оно используется для выделения услуги метаданных из потока метаданных.

metadata_locator_record_flag: Флаг metadata_locator_record_flag – это 1-битовое поле, которое, когда оно установлено в '1', указывает, что связанные с ним метаданные доступны в местоположении вне потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, определенном в поле metadata_locator_record.

MPEG_carriage_flags: Флаги MPEG_carriage_flags – это 2-битовое поле, которое определяет, передается ли поток метаданных, содержащих связанную с ним услугу метаданных, в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, и, если да, то передаются ли метаданные в Транспортном или Программном потоке. Кодирование этого поля определяется в таблице 2-85.

Таблица 2-85 – Флаги MPEG_carrier_flags

Значение	Описание
0	Передача в том же Транспортном потоке, в котором передается этот дескриптор указателя метаданных.
1	Передача в Транспортном потоке, отличном от того, в котором передается этот дескриптор указателя метаданных.
2	Передача в Программном потоке. Это может быть тот же Программный поток, в котором передается этот дескриптор указателя метаданных, или может быть другой Программный поток.
3	Ничего из вышеназванного.

metadata_locator_record_length: Поле metadata_locator_record_length – это 8-битовое поле, которое определяет количество байтов metadata_locator_record_bytes, которые следуют сразу после него. Это поле не должно быть закодировано со значением '0'.

metadata_locator_record_byte: Байт metadata_locator_record_byte – это участок строки из одного или нескольких последовательных байтов, которая формирует запись местоположения метаданных. Эта запись определяет одно или несколько местоположений вне потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Формат записи о местоположении метаданных определяется приложением метаданных, информация о котором передается в поле metadata_application_format. Эта запись может, например, содержать сведения об URL интернета, которые определяют, где могут находиться эти метаданные, возможно, в дополнение к их местоположению(ям) в Транспортном потоке. Если флаги MPEG_carriage_flags кодируются со значением '0', '1' или '2', и присутствует запись местоположения метаданных, то это говорит о наличии дополнительных местоположений для тех же самых метаданных.

program_number: Поле program_number – это 16-битовое поле, которое указывает номер программы MPEG-2 в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. в котором передаются связанные с ним метаданные. Если поле MPEG_carriage_flags имеет значение '0', то Транспортный поток является текущим, а если поле MPEG_carriage_flags имеет значение '1', то программой являются поля transport_stream_location и transport_stream_id Транспортного потока.

transport_stream_location: Поле transport_stream_location – это 16-битовое поле, которое определено в конфиденциальном порядке. Например, это поле может использоваться приложениями для передачи идентификатора original_network_id, определенного ETSI.

transport_stream_id: Поле transport_stream_id – это 16-битовое поле, которое указывает Транспортный поток, в котором передаются связанные с ним метаданные.

private_data_byte: Поле private_data_byte – это 8-битовое поле. Поле private_data_byte представляет собой данные, формат которых определен в конфиденциальном порядке. Эти байты могут использоваться для предоставления дополнительной информации, которая представляется необходимой.

2.6.60 Дескриптор метаданных

Дескриптор метаданных определяет параметры услуги метаданных, передаваемой в потоках MPEG-2 TS или PS. Для элементарного потока, в котором передается услуга метаданных, дескриптор MPEG-2 TS включается в таблицу PMT в цепочке дескриптора. Дескриптор определяет формат связанных с ним метаданных и содержит значение идентификатора metadata_service_id для идентификации услуги метаданных, к которой применяется этот дескриптор метаданных. При необходимости, дескриптор может содержать информацию, требуемую для идентификации услуги метаданных в наборе метаданных, передаваемых в карусели DSM-CC. Дополнительно может передаваться конфиденциальная информация, определяемая форматом приложения метаданных.

Дескриптор метаданных также сообщает о том, требуется ли конфигурирование декодера, и способен ли он содержать байты конфигурации декодера, но это практично только в том случае, когда число этих байтов мало. Если информация о конфигурации декодера чересчур объемна для того, чтобы ее передавать в дескрипторе, она должна содержаться в услуге метаданных. Она может находиться внутри самой услуги метаданных или внутри другой услуги метаданных в пределах той же программы. Идентификация услуги метаданных, которая содержит информацию о конфигурации декодера, выполняется при помощи дескриптора метаданных. Если для передачи информации о конфигурации декодера используется карусель DSM-CC, то может быть представлена информация о том, как получить из карусели информацию о конфигурации декодера.

Таблица 2-86 – Дескриптор метаданных

Синтаксис	Количество битов	Мнемоника
Metadata_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
metadata_application_format	**16**	**uimsbf**
if (metadata_application_format == 0xFFFF) {		
metadata_application_format_identifier	**32**	**uimsbf**
}		
metadata_format	**8**	**uimsbf**
if (metadata_format== 0xFF){		
metadata_format_identifier	**32**	**uimsbf**
}		
metadata_service_id	**8**	**uimsbf**
decoder_config_flags	**3**	**bslbf**
DSM-CC_flag	**1**	**bslbf**
зарезервировано	**4**	**bslbf**
if (DSM-CC_flag == '1'){		

Синтаксис	Количество битов	Мнемоника
service_identification_length	**8**	**uimsbf**
for(i=0; i<service_identification_length; i++) {		
service_identification_record_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '001') {		
decoder_config_length	**8**	**uimsbf**
for(i=0; i<decoder_config_length; i++) {		
decoder_config_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '011') {		
dec_config_identification_record_length	**8**	**uimsbf**
for(i=0;i<dec_config_id_record_length;i++) {		
dec_config_identification_record_byte	**8**	**bslbf**
}		
}		
if (decoder_config_flags == '100') {		
decoder_config_metadata_service_id	**8**	**uimsbf**
}		
if (decoder_config_flags == '101'|'110') {		
reserved_data_length	**8**	**uimsbf**
for(i=0;i<reserved_data_length;i++) {		
зарезервировано	**8**	**bslbf**
}		
}		
for (i=0; i<N;i++) {		
private_data_byte	**8**	**bslbf**
}		
}		

2.6.61 Семантические определения полей в дескрипторе метаданных

metadata_application_format: Поле metadata_application_format – это 16-битовое поле, которое в этом дескрипторе ответственно за определение использования, синтаксиса и семантики записи service_identification_record и любых других полей, которые определены в конфиденциальном порядке. Кодирование этого поля определяется в таблице 2-81.

metadata_application_format_identifier: Кодирование этого поля определяется в § 2.6.57.

metadata_format: Кодирование этого поля определяется в § 2.6.59.

metadata_format_identifier: Кодирование этого поля определяется в § 2.6.59.

metadata_service_id: Это 8-битовое поле указывает услугу метаданных, к которой относится данный дескриптор метаданных.

decoder_config_flags: Поле decoder_config_flags – это 3-битовое поле, которое указывает, передается ли информация конфигурации декодера, и как она передается.

Таблица 2-87 – Флаги decoder_config_flags

Значение	Описание
000	Конфигурации декодера не требуется.
001	Информация о конфигурации декодера передается в этом дескрипторе в поле decoder_config_byte.
010	Информация о конфигурации декодера передается в той же услуге метаданных, к которой применяется данный дескриптор метаданных.
011	Информация о конфигурации декодера передается в карусели DSM-CC. Это значение должно использоваться только в том случае, когда услуга метаданных, к которой применяется данный дескриптор метаданных, использует тот же тип карусели DSM-CC.
100	Информация о конфигурации декодера передается в другой услуге метаданных в той же программе, указанной в поле decoder_config_metadata_service_id данного дескриптора метаданных.
101, 110	Зарезервированы.
111	Определено в конфиденциальном порядке.

DSM-CC_flag: Это однобитовый флаг, который установлен в '1', если поток, с которым связан этот дескриптор, передается в данных, соответствующих ИСО/МЭК 13818-6, или в объекте карусельного типа.

ПРИМЕЧАНИЕ 1. – Использование объекта или данных карусельного типа указывается применяемым значением типа потока для данного потока метаданных.

service_identification_length: Это поле определяет число байтов service_identification_record_byte, которые следуют сразу после него.

service_identification_record_byte: Этот байт представляет собой участок строки из одного или нескольких последовательных байтов, которая определяет запись service_identification_record. Эта запись содержит информацию о получении услуги метаданных из карусели DSM-CC. Формат записи о местоположении метаданных определяется приложением, указанным форматом приложения метаданных. Когда используется объект DSM-CC карусельного типа, эта запись может, например, включать в себя уникальный идентификатор объекта (IOP:IOR() из § 11.3.1 и § 5.7.2.3 стандарта ИСО/МЭК 13818-6 DSM-CC) для услуги метаданных. Аналогично, в случае использования данных карусельного типа DSM-CC, эта запись может, например, предоставлять идентификатор услуги метаданных transaction_id и идентификатор услуги метаданных module_id.

decoder_config_length: Это поле определяет число байтов decoder_config_byte, которые следуют сразу после него.

decoder_config_byte: Эти байты содержат информацию о конфигурации декодера. Эта последовательность байтов содержит информацию о конфигурации, необходимую приемнику для декодирования этой услуги. Предполагается, что передача этой информации в дескрипторе метаданных используется только, если объем информации о конфигурации декодера очень мал.

decoder_config_DSM-CC_id: Это идентификатор загрузки информации о конфигурации декодера, когда она передается в данных карусельного типа DSM-CC, или это идентификатор объекта информации о конфигурации декодера, если она передается в данных карусельного типа DSM-CC.

ПРИМЕЧАНИЕ 2. – Использование объекта или данных карусельного типа указывается применяемым значением типа потока для данного потока метаданных.

dec_config_identification_record_length: Это поле определяет количество следующих сразу после него байтов dec_config_identification_record_bytes.

dec_config_identification_record_byte: Этот байт представляет собой участок строки из одного или нескольких последовательных байтов, которая определяет запись dec_config_identification_record. Эта запись определяет, как получить из карусели DSM-CC требуемую информацию о конфигурации декодера. Формат записи о местоположении метаданных определяется форматом приложения метаданных. Когда используется объект DSM-CC карусельного типа, эта запись может, например, включать в себя уникальный идентификатор объекта (IOP:IOR() из § 11.3.1 и § 5.7.2.3 стандарта ИСО/МЭК 13818-6 DSM-CC) для конфигурации декодера. Аналогично, в случае использования данных карусельного типа DSM-CC, эта запись может, например, предоставлять идентификатор конфигурации декодера transaction_id и идентификатор конфигурации декодера module_id.

decoder_config_metadata_service_id: Это значение идентификатора metadata_service_id, который присвоен услуге метаданных, содержащей информацию о конфигурации декодера. Услуга метаданных, указанная идентификатором decoder_config_metadata_service_id, и услуга метаданных, которая использует информацию о конфигурации декодера, должны находиться водной программе. Отсюда, в Транспортном потоке дескрипторы метаданных для обеих этих услуг метаданных должны находиться в одной таблице PMT. Дескриптор метаданных услуги метаданных, указанный в идентификаторе decoder_config_metadata_service_id, должен иметь поле decoder_config_flag со значением '001', '010' или '011'.

reserved_data_length: Это поле определяет число зарезервированных байтов, которые следуют сразу после него.

private_data_byte: Поле private_data_byte – это 8-битовое поле. Поля private_data_byte представляют собой данные, формат которых определен в конфиденциальном порядке. Эти байты могут использоваться для предоставления дополнительной информации, которая представляется необходимой.

2.6.62 Дескриптор метаданных декодера STD

Этот дескриптор определяет параметры модели декодера STD (определенной в § 2.12.10) для обработки потока метаданных, с которым связан этот дескриптор.

Таблица 2-88 – Дескриптор метаданных декодера STD

Синтаксис	Количество битов	Мнемоника
Metadata_STD_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
зарезервировано	**2**	**bslbf**
metadata_input_leak_rate	**22**	**uimsbf**
зарезервировано	**2**	**bslbf**
metadata_buffer_size	**22**	**uimsbf**
зарезервировано	**2**	**bslbf**
metadata_output_leak_rate	**22**	**uimsbf**
}		

2.6.63 Семантические определения полей в дескрипторе метаданных декодера STD

metadata_input_leak_rate: Поле metadata_input_leak_rate – это 22-битовое поле, которое определяет в модели T-STD скорость утечки для соответствующего потока метаданных из буфера TB_n в буфер B_n. Скорость утечки определяется в единицах по 400 бит/с. Для метаданных, передаваемых в Программном потоке, кодирование поля metadata_input_leak_rate не определено, поскольку скорость поступления данных в буфер B_n равна скорости Программного потока.

metadata_buffer_size: Поле metadata_buffer_size – это 22-битовое поле, которое определяет в модели STD размер буфера B_n для связанного с ним потока метаданных. Размер буфера B_n определяется в единицах по 1024 байта.

metadata_output_leak_rate: Поле metadata_output_leak_rate – это 22-битовое поле, которое в модели STD для связанной с ним услуги метаданных определяет скорость утечки информации из буфера B_n в декодер. Скорость утечки определяется в единицах по 400 бит/с. Для синхронно передаваемых потоков метаданных (тип потока 0x15 или 0x19), модули доступа метаданных моментально удаляются из буфера B_n под управлением меток времени PTS, в этом случае кодирование поля metadata_output_leak_rate не определяется.

2.6.64 Дескриптор видеопотока AVC

Для видеопотоков, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, дескриптор видеопотока AVC содержит основную информацию, требуемую для идентификации таких параметров кодирования соответствующего видеопотока AVC, как профиль и параметры уровня, включенные в SPS видеопотока AVC.

Дескриптор видеопотока AVC также сообщает о присутствии в видеопотоке AVC неподвижных изображений и 24-часовых изображений AVC. Если данный дескриптор не включен в таблицу PMT для видеопотока AVC в Транспортном потоке или в PSM для видеопотока AVC в Программном потоке, то такой видеопоток не должен содержать неподвижных или 24-часовых изображений AVC (см. таблицу 2-89.)

Таблица 2-89 – Дескриптор видеопотока AVC

Синтаксис	Количество битов	Мнемоника
AVC_video_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
profile_idc	**8**	**uimsbf**
constraint_set0_flag	**1**	**bslbf**
constraint_set1_flag	**1**	**bslbf**
constraint_set2_flag	**1**	**bslbf**
AVC_compatible_flags	**5**	**bslbf**
level_idc	**8**	**uimsbf**
AVC_still_present	**1**	**bslbf**
AVC_24_hour_picture_flag	**1**	**bslbf**
Зарезервировано	**6**	**bslbf**
}		

2.6.65 Семантическое определение полей в дескрипторе видеопотока AVC

profile_idc, constraint_set0_flag, **constraint_set1_flag**, **constraint_set2_flag, AVC_compatible_flags и level_idc** – Эти поля, за исключением поля AVC_compatible_flags, должны кодироваться в соответствии с семантикой, определенной в Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Семантика поля AVC_compatible_flags в точности соответствует семантике полей, определенной для 5 битов между флагом constraint_set2 и полем level_idc в наборе параметров последовательности, описанном в Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. В целом видеопоток AVC, с которым связан дескриптор AVC, должен соответствовать профилю, уровню и ограничениям, задаваемым данными полями.

ПРИМЕЧАНИЕ. – В одной или нескольких последовательностях видеопотока AVC уровень может быть ниже указанного в дескрипторе видеопотока AVC, так же как и профиль может быть подмножеством профиля, заданного дескриптором видеопотока AVC. Однако во всем видеопотоке AVC могут использоваться только инструменты, включенные в профиль, заданный дескриптором, если он присутствует. Например, если задан основной профиль, то в некоторых последовательностях может использоваться базовый профиль, но только с использованием инструментов, имеющихся в основном профиле. Если наборы параметров последовательности в видеопотоке AVC задаются разными профилями, и нет никаких дополнительных ограничений, то может потребоваться определение того, какой профиль, если он есть, соответствует потоку в целом. Если дескриптор видеопотока AVC относится к видеопотоку AVC, который соответствует нескольким профилям, то этот видеопоток должен быть разбит на два или более подпотоков так, чтобы их дескрипторы определялись одним профилем для каждого подпотока.

AVC_still_present – Это 1-битовое поле, которое, когда оно установлено в '1', указывает, что видеопоток AVC может содержать неподвижные изображения AVC. Если его значение равно '0', то соответствующий видеопоток AVC не должен содержать неподвижных изображений AVC.

AVC_24_hour_picture_flag – Это 1-битовое поле, которое, когда оно установлено в '1', указывает что видеопоток AVC может содержать 24-часовые изображения AVC. Определение 24-часового изображения AVC см. в § 2.1.2. Если этот флаг установлен в '0', то соответствующий видеопоток AVC не должен содержать 24-часовых изображений AVC.

2.6.66 Дескриптор синхронизации AVC и HRD

Дескриптор синхронизации AVC и HRD предоставляет параметры синхронизации и HRD для соответствующего видеопотока AVC. Для каждого видеопотока AVC, передаваемого в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, дескриптор синхронизации AVC и HRD должен быть включен в таблицу PMT или в карту PSM, если он присутствует в программном потоке, если только видеопоток AVC не содержит параметров VUI (возможности использования видеоинформации) с флагом timing_info_present_flag, установленным в '1':

- для каждого изображения IDR; и
- для каждого изображения, связанного с точкой восстановления сообщения SEI.

Отсутствие дескриптора синхронизации AVC и HRD в PMT для видеопотока AVC свидетельствует об использовании в декодере T-STD для передачи данных из MB_n в EB_n метода утечки, определенного в § 2.14.3.1, но об этом также может говорить флаг hrd_management_valid_flag в дескрипторе синхронизации AVC и HRD, установленный в '0'. Если скорость передачи информации в буфер EB_n может быть определена в параметрах HRD, содержащихся в видеопотоке AVC, и если эта скорость передачи используется в T-STD при передаче сигнала из MB_n в EB_n, то дескриптор синхронизации AVC и HRD с флагом hrd_management_valid_flag, установленным в '1', должен быть включен в таблицу PMT для данного видеопотока AVC (см. таблицу 2-90.)

Таблица 2-90 – Дескриптор синхронизации AVC и HRD

Синтаксис	Количество битов	Мнемоника
Дескриптор синхронизации AVC и HRD () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
hrd_management_valid_flag	**1**	**bslbf**
Зарезервировано	**6**	**bslbf**
picture_and_timing_info_present	**1**	**bslbf**
if (picture_and_timing_info_present) {		
90kHz_flag	**1**	**bslbf**
Зарезервировано	**7**	**bslbf**
if (90kHz_flag = = '0') {		
N	**32**	**uimsbf**
K	**32**	**uimsbf**
}		
num_units_in_tick	**32**	**uimsbf**

Синтаксис	Количество битов	Мнемоника
}		
fixed_frame_rate_flag	**1**	**bslbf**
temporal_poc_flag	**1**	**bslbf**
picture_to_display_conversion_flag	**1**	**bslbf**
Зарезервировано	**5**	**bslbf**
}		

2.6.67 Семантическое определение полей в дескрипторе синхронизации AVC и HRD

hrd_management_valid_flag – Это 1-битовое поле определено только для использования в Транспортных потоках.

Если дескриптор синхронизации AVC и HRD связан с видеопотоком AVC, передаваемым в составе Транспортного потока, то применяется следующее правило. Если флаг hrd_management_valid_flag установлен в '1', то в соответствующем видеопотоке AVC должны присутствовать сообщения дополнительной расширенной информации (SEI) периода буферизации и SEI синхронизации изображения, определенные в Приложении C Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Эти сообщения SEI периода буферизации должны содержать кодированные значения полей initial_cpb_removal_delay и initial_cpb_removal_delay_offset для абстрактного уровня сети (NAL) эталонного декодера HRD. Если флаг hrd_management_valid_flag установлен в '1', то передача каждого байта из буфера MB_n в буфер EB_n в декодере T-STD должна выполняться в соответствии с расписанием доставки данного байта в буфер кодированного изображения (CPB) декодера HRD NAL, как определено в кодированных значениях полей initial_cpb_removal_delay и initial_cpb_removal_delay_offset для SchedSelIdx = cpb_cnt_minus1. Если флаг hrd_management_valid_flag установлен в '0', то в T-STD для передачи данных из буфера MB_n в буфер EB_n должен использоваться метод утечки, описанный в § 2.14.3.

Если дескриптор синхронизации AVC и HRD связан с видеопотоком AVC, передаваемым в составе программного потока, то значение флага hrd_management_valid_flag не определено.

picture_and_timing_info_present – Это 1-битовое поле, которое, когда оно установлено в '1', указывает, что в данный дескриптор включены флаг 90kHz_flag и параметры для точного преобразования в значения системных часов с частотой 90 кГц.

90kHz_flag, N, K – Если флаг 90kHz_flag установлен в '1', то это означает, что частота временной базы видеопотока AVC равна 90 кГц. Для видеопотока AVC частота временной базы AVC определяется параметром AVC time_scale в параметрах VUI, как описано в Приложении E Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Соотношение между параметром AVC time_scale и STC определяется в данном дескрипторе параметрами N и K следующим образом.

$$time_scale = \frac{(N \times system_clock_frequency)}{K},$$

где time_scale означает точную частоту временной базы AVC, причем K больше или равен N.

Если флаг 90kHz_flag установлен в '1', то N равно 1, а K равно 300. Если флаг 90kHz_flag установлен в '0', то значения N и K задаются значениями полей N и K.

ПРИМЕЧАНИЕ 1. – Это позволяет преобразовывать время, выраженное в единицах временной шкалы time_scale, во время, выраженное в единицах по 90 кГц, как требуется для вычисления меток времени PTS и DTS, например, в декодерах для блоков доступа AVC, для которых нет ни PTS, ни DTS, закодированных в заголовке PES.

num_units_in_tick – кодируется точно таким же образом, как и поле num_units_in_tick в параметрах VUI, описанных в Приложении E Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Информация, содержащаяся в этом поле, должна применяться ко всему видеопотоку AVC, с которым связан дескриптор синхронизации AVC и HRD.

fixed_frame_rate_flag – кодируется точно таким же образом, как и флаг fixed_frame_rate_flag в параметрах VUI, описанных в Приложении E Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Если этот флаг установлен в '1', то это означает, что в пределах соответствующего видеопотока кодированная частота кадров постоянна. Если он установлен в '0', то в данном дескрипторе не задается никакой информации о частоте кадров соответствующего видеопотока AVC.

temporal_poc_flag – если флаг temporal_poc_flag установлен в '1' и флаг fixed_frame_rate_flag тоже установлен в '1', то в соответствующем видеопотоке AVC передается информация счетчика запросов изображений (POC) (PicOrderCnt), в которой изображения подсчитываются в единицах $\Delta t_{fi,dpb}(n)$, где $\Delta t_{fi,dpb}(n)$ определяется в уравнении E-10 Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Если флаг temporal_poc_flag установлен в '0', то не передается никакой информации, относящейся к потенциальным соотношениям между данными POC в видеопотоке AVC и временем.

ПРИМЕЧАНИЕ 2. – Это уменьшает накладные расходы, требуемые для синхронизации сигнала в каждом модуле доступа. Для модуля доступа, для которого не передаются точные значения PTS/DTS, могут быть вычислены эффективные метки времени PTS и DTS. Если разница между значениями PTS в текущем и следующем изображении больше, чем $2 \times \Delta t_{fi,dpb}$ (или больше чем $\Delta t_{fi,dpb}$, если флаг *frame_mbs_only_flag* равен 1), то требуется повторить последних по времени поля соответствующего полукадра или кадра.

picture_to_display_conversion_flag – Если это 1-битовое поле установлено в '1', то это значит, что информация отображения для кодированных изображений в соответствующем видеопотоке AVC может передаваться путем задания поля pic_struct в сообщениях SEI picture_timing (см. Приложение D Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10) и/или путем задания значения PicOrderCnt счетчика запросов изображения (POC), где изображения подсчитываются в единицах $\Delta t_{fi,dpb}(n)$ (см. также семантику флага temporal_poc_flag), так что информация синхронизации для следующего модуля доступа AVC может быть получена из изображения, являющегося предыдущим по порядку декодирования или воспроизведения.

Если флаг picture_to_display_conversion_mode_flag установлен в '0', то сообщения синхронизации изображений SEI, если они присутствуют в видеопотоке AVC, не содержат поля pic_struct и, следовательно, флаг pic_struct_present_flag должен быть установлен в '0' в параметрах VUI видеопотока AVC.

2.6.68 Дескриптор аудиопотока MPEG-2 AAC

Для отдельных потоков, соответствующих ИСО/МЭК 13818-7, передаваемых непосредственно в пакетах PES, дескриптор аудиопотока MPEG-2 AAC, определенный в таблице 2-91, предоставляет базовую информацию для идентификации параметров кодирования таких элементарных аудиопотоков.

Таблица 2-91 – Дескриптор аудиопотока MPEG-2 AAC

Синтаксис	Количество битов	Мнемоника
MPEG-2_AAC_audio_descriptor () {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
MPEG-2_AAC_profile	**8**	**uimsbf**
MPEG-2_AAC_channel_configuration	**8**	**uimsbf**
MPEG-2_AAC_additional_information	**8**	**uimsbf**
}		

2.6.69 Семантические определения полей в дескрипторе аудиопотока MPEG-2 AAC

MPEG-2_AAC_profile – Это 8-битовое поле указывает профиль AAC в соответствии с индексом из таблицы 31 стандарта ИСО/МЭК 13818-7:2006.

MPEG-2_AAC_channel_configuration – Это 8-битовое поле указывает количество и конфигурацию аудиоканалов, представленных декодером AAC слушателю определенной программы. Значения в диапазоне от 1 до 6 указывают количество и конфигурацию аудиоканалов, обозначенных в таблице 42 стандарта ИСО/МЭК 13818-7:2006 как "номер индекса по умолчанию". Все остальные значения указывают, что количество и конфигурация аудиоканалов не определены.

MPEG-2_AAC_additional_information – Это 8-битовое поле указывает, введена ли в двоичный поток AAC информация о расширении полосы пропускания, в соответствии с таблицей 2-92.

Таблица 2-92 – Значения поля MPEG-2_AAC_additional_information

Значение	Описание
0x00	Данные AAC, соответствующие ИСО/МЭК 13818-7:2006
0x01	Данные AAC с информацией о расширении полосы пропускания, соответствующие ИСО/МЭК 13818-7:2006
0x02-0xFF	Зарезервированы

2.7 Ограничения семантики мультиплексированного потока

2.7.1 Частота кодирования эталонного времени системы

Программный поток должен быть сформирован таким образом, что интервал времени между байтами, содержащими последний бит полей system_clock_reference_base в последовательных блок-пакетах, должен быть меньше или равен 0,7 с. Следовательно:

$$|t(i) - t(i')| \le 0{,}7 \text{ с}$$

для всех i и i′, где i и i′ – индексы байтов, содержащих последний бит последовательных полей system_clock_reference_base.

2.7.2 Частота кодирования эталонного времени программы

Транспортный поток должен быть сформирован таким образом, что для каждой программы интервал времени между байтами, содержащими последний бит полей program_clock_reference_base, в последовательных появлениях PCR в пакетах Транспортного потока с идентификатором PCR_PID должен быть меньше или равен 0,1 с. Следовательно:

$$|t(i) - t(i')| \le 0{,}1\,\text{с}$$

для всех i и i′, где i и i′ – индексы байтов, содержащих последний бит последовательных полей program_clock_reference_base в пакетах Транспортного потока с идентификатором PCR_PID для каждой программы.

Для осуществления фазовой синхронизации и экстраполяции значений времени доставки байтов, необходимо иметь, как минимум, два (2) значения эталонного времени программы (PCR), находящихся в Транспортном потоке после указанного идентификатора PCR_PID, между последовательными случаями нарушения непрерывности PCR (см. § 2.4.3.4).

2.7.3 Частота кодирования эталонного времени элементарного потока

Программный поток и Транспортный поток должны быть сформированы таким образом, при котором, если поле эталонного времени элементарного потока закодировано в любых пакетах PES, содержащих информацию данного элементарного потока, то интервал времени в декодере PES_STD между байтами, содержащими последний бит последовательных полей ESCR_base, должен быть меньше или равен 0,7 с. Кодирования ESCR в потоках пакетов PES требуется выполнять с тем же интервалом. Следовательно:

$$|t(i) - t(i')| \le 0{,}7\,\text{с}$$

для всех i и i′, где i и i′ – индексы байтов, содержащих последние биты последовательных полей ESCR_base.

ПРИМЕЧАНИЕ. – Кодирование полей эталонного времени элементарного потока является дополнительным; кодировать их не обязательно. Однако если они кодируются, то применяется это ограничение.

2.7.4 Частота кодирования меток времени воспроизведения

Программный поток и Транспортный поток должны быть сформированы таким образом, чтобы максимальная разница между кодированными метками времени воспроизведения для каждого элементарного видео или аудиопотока составляла 0,7 с. Следовательно:

$$|tp_n(k) - tp_n(k'')| \le 0{,}7\,\text{с}$$

для всех n, k и k″, которые удовлетворяют условиям:

- $P_n(k)$ и $P_n(k'')$ – это модули представления, для которых кодируются метки времени воспроизведения;
- k и k″ выбираются таким образом, чтобы не могло появиться модуля представления $P_n(k')$ с кодированной меткой времени воспроизведения и с $k < k' < k''$; и
- в элементарном потоке "n" нет нарушения непрерывности декодирования между модулями $P_n(k)$ и $P_n(k'')$.

Ограничение в 0,7 с не применяется в случае:

- неподвижных изображений, как описано в § 2.1;
- неподвижных изображений AVC;
- модулей доступа AVC с очень низкой частотой кадров, где время воспроизведения последовательных модулей доступа отличается более чем на 0,7 с. В этом частном случае параметры VUI num_units_in_tick и time_scale должны присутствовать или в видеопотоке AVC, или в дескрипторе синхронизации AVC и HRD, связанном с видеопотоком AVC.

ПРИМЕЧАНИЕ. – Время воспроизведения модуля доступа AVC эквивалентно выходному времени DPB $t_{o,dpb}(n)$, определенному в Приложении C Рекомендации МСЭ-T H.264 | ИСО/МЭК 14496-10.

2.7.5 Условное кодирование меток времени

Для каждого элементарного потока Программного потока или Транспортного потока, метка времени воспроизведения (PTS) должна быть закодирована для первого модуля доступа.

Нарушение непрерывности декодирования в начале модуля доступа $A_n(j)$ элементарного потока "n" существует, если время декодирования $td_n(j)$ этого модуля доступа больше наибольшего значения, допустимого с учетом определенных допусков для частоты системных часов (system_clock_frequency). Для видеоданных, за исключением случая, когда состояние "хитрого" режима = "true", или когда флаг low_delay = '1', нарушение непрерывности допускается только в начале видеопоследовательности. Если нарушение непрерывности декодирования существует в каком либо элементарном видео или аудиопотоке в Транспортном или Программном потоке, то метка PTS должна кодироваться для первого модуля доступа после каждого случая нарушения непрерывности декодирования, за исключением случая, когда состояние "хитрого" режима = "true".

Когда флаг low_delay = '1', метка PTS должна кодироваться для первого модуля доступа после освобождения буфера EB_n или B_n.

Метка PTS может присутствовать в заголовке пакета PES элементарного видео или аудиопотока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, только если в этом пакете PES присутствует первый байт стартового кода кадра или первый байт модуля доступа аудиопотока.

Метка времени декодирования decoding_timestamp (DTS) должна появляться в заголовке пакета PES, если, и только, если выполняются два следующих условия:

- в заголовке пакета PES присутствует метка PTS;
- время декодирования отличается от времени воспроизведения.

Явные значения PTS и DTS не должны кодироваться в заголовке PES для каждого 24-часового изображения AVC. Декодеры должны определять время воспроизведения таких модулей доступа AVC, зная параметры, передаваемые в видеопотоке AVC. Следовательно, каждый видеопоток AVC, содержащий одно или несколько 24-часовых изображений AVC:

- либо должен содержать сообщения синхронизации изображения SEI с кодированными значениями полей cpb_removal_delay и dpb_output_delay;
- либо должен содержать параметры VUI с флагом fixed_frame_rate_flag, установленным в '1', а также информацию (PicOrderCnt) счетчика запросов изображений (POC), в которой изображения подсчитываются в единицах $\Delta t_{fi,dpb}(n)$, где $\Delta t_{fi,dpb}(n)$ определено в уравнении E-10 Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10.

ПРИМЕЧАНИЕ 1. – Второе требование возникает, если дескриптор синхронизации AVC и HRD связан с видеопотоком AVC, в котором флаг fixed_frame_rate_flag установлен в '1', и флаг temporal_poc_flag тоже установлен в '1'.

Нижесказанное применимо к модулям доступа AVC в видеопотоке AVC, передаваемом в составе потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Для каждого модуля доступа AVC, который не воспроизводит 24-часовых изображений AVC, должен быть сформирован заголовок PES с кодированной меткой PTS, а также, если применимо, меткой DTS, за исключением случая, когда выполняются все перечисленные ниже условия:

- В видеопоследовательности AVC присутствуют следующие сообщения SEI, о которых сообщают соответствующие параметры VUI:
  - a) сообщения SEI синхронизации изображений, содержащие значения параметров cpb_removal_delay и dpb_output_delay; и
  - b) сообщения SEI периода буферизации, содержащие значения параметров initial_cpb_removal_delay и initial_cpb_removal_delay_offset.

  ПРИМЕЧАНИЕ 2. – Если в видеопоследовательности AVC присутствуют сообщения SEI синхронизации кадров, то они представлены для каждого модуля доступа AVC, как того требует Рекомендация МСЭ-Т H.264 | ИСО/МЭК 14496-10. Если в видеопоследовательности AVC присутствуют сообщения SEI периода буферизации, то они должны быть представлены для каждого модуля доступа IDR и для каждого модуля доступа, связанного с точкой восстановления сообщения SEI, как того требует Рекомендация МСЭ-Т H.264 | ИСО/МЭК 14496-10.
- Дескриптор синхронизации AVC и HRD связан с видеопотоком AVC, и в этом дескрипторе флаги fixed_frame_rate_flag и temporal_poc_flag установлены в '1'.
- Дескриптор синхронизации AVC и HRD связан с видеопотоком AVC, и в этом дескрипторе флаг fixed_frame_rate_flag установлен в '1', флаг picture_to_display_conversion_flag установлен в '1', флаг temporal_poc_flag установлен в '0', и в видеопоследовательности AVC присутствуют сообщения SEI синхронизации изображений с полем pic_struct.

ПРИМЕЧАНИЕ 3. – В этом частном случае поле pic_struct используется для определения последующих значений PTS.

- Дескриптор синхронизации AVC и HRD связан с видеопотоком AVC, и в этом дескрипторе флаг fixed_frame_rate_flag установлен в '1', флаг temporal_poc_flag установлен в '0' и флаг picture_to_display_conversion_flag также установлен в '0'.

  ПРИМЕЧАНИЕ 4. – В этом случае информация POC в видеопотоке AVC используется для определения последующих значений PTS.

2.7.6 Ограничения синхронизации при масштабируемом кодировании

Если аудиопоследовательность кодируется с использованием расширения битового потока, определенного в ИСО/МЭК 13818-3, то соответствующие модули декодирования/воспроизведения в двух уровнях имеют идентичные значения PTS.

Если видеопоследовательность кодируется как расширение SNR другой последовательности, как определено в § 7.8 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, то набор значений времени воспроизведения для обеих последовательностей должен быть одинаковым.

Если видеопоследовательность кодируется как два логических фрагмента, как определено в § 7.10 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, то набор значений времени воспроизведения для обоих фрагментов должен быть одинаковым.

Если видеопоследовательность кодируется как пространственное масштабируемое расширение другой последовательности, как определено в § 7.7 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, то должны применяться следующие правила:

- Если обе последовательности имеют одну и ту же частоту кадров, то набор значений времени воспроизведения для них должен быть одинаковым.

  ПРИМЕЧАНИЕ. – Это не означает, что тип кодирования изображения одинаков для обоих уровней.

- Если последовательности имеют разную частоту кадров, то набор значений времени воспроизведения должен быть таким, чтобы в обеих последовательностях совпадало максимально возможное количество значений времени воспроизведения.
- Изображение, относительно которого выполняется пространственное прогнозирование, должно быть одним из следующих:
  - совпадающее по времени или последнее по времени декодированное изображение более низкого уровня;
  - совпадающее по времени или последнее по времени декодированное изображение более низкого уровня, являющееся I- или P-изображением;
  - предпоследнее по времени декодированное изображение более низкого уровня, являющееся I- или P-изображением, и при этом на более низком уровне нет параметра low_delay, установленного в '1".

Если видеопоследовательность кодируется как временное масштабируемое расширение другой последовательности, как определено в § 7.9 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, то в качестве опорных могут использоваться следующие изображения более низкого уровня. Значения времени рассчитываются относительно времени воспроизведения:

- совпадающее по времени или последнее по времени воспроизведенное изображение более низкого уровня;
- следующее изображение более низкого уровня, которое должно быть воспроизведено.

2.7.7 Частота кодирования P-STD_buffer_size в заголовках пакета PES

В Программном потоке поля P-STD_buffer_scale и P-STD_buffer_size должны присутствовать в первом пакете PES каждого элементарного потока и появляться снова всякий раз, когда их значение меняется. Они могут также присутствовать в любом другом пакете PES.

2.7.8 Кодирование системного заголовка в Программном потоке

В Программном потоке системный заголовок может присутствовать в любом блок-пакете, который следует сразу после заголовка блок-пакета. Системный заголовок должен присутствовать в первом блок-пакете Программного потока. Значения, закодированные во всех системных заголовках в Программном потоке, должны быть идентичными.

2.7.9 Программный поток с ограниченными системными параметрами

Программный поток является "Потоком с ограниченными системными параметрами" (CSPS), если он вписывается в границы, определенные в настоящем подразделе. Программные потоки не ограничиваются пределами, определенными потоком CSPS. Поток CSPS может быть идентифицирован при помощи флага CSPS_flag, определенного в системном заголовке в § 2.5.3.5. Поток CSPS представляет собой подмножество всех возможных Программных потоков.

Скорость передачи пакетов

В потоке CSPS максимальная скорость, с которой пакеты должен поступать на вход декодера P-STD составляет 300 пакетов в секунду, если значение, закодированное в поле rate_bound (см. § 2.5.3.6) меньше или равно 4 500 000 бит/с, если флаг packet_rate_restriction_flag установлен в ‘1', и меньше или равно 2 000 000 бит/с, если флаг packet_rate_restriction_flag установлен в '0'. Для более высоких скоростей скорость передачи пакетов в потоке CSPS ограничивается линейной зависимостью от значения, закодированного в поле rate_bound.

В частности, для всех блок-пакетов Программного потока, когда флаг packet_rate_restriction_flag (см. § 2.5.3.5) установлен в '1',

$$NP \le \left(t(i') - t(i')\right) \times 300 \times \max\left[1, \frac{R_{\max}}{4{,}5 \times 10^6}\right] \qquad (2\text{-}27)$$

и если флаг packet_rate_restriction_flag установлен в '0'

$$NP \le \left(t(i') - t(i')\right) \times 300 \times \max\left[1, \frac{R_{\max}}{2{,}5 \times 10^6}\right], \qquad (2\text{-}28)$$

где:

$$R_{\max} = 8 \times 50 \times rate_bound \quad бит/с \qquad (2\text{-}29)$$

NP количество префиксов packet_start_code_prefixe и кодов system_header_start_code между смежными кодами pack_start_code или между последним кодом pack_start_code и кодом MPEG_program_end_code, определенными в таблице 2-37, семантика которых определена в § 2.5.3.2.

t(i) - время (измеренное в секундах), закодированное в SCR для блок-пакета p.

t(i′) - время (измеренное в секундах), закодированное в SCR для блок-пакета p + 1, который следует сразу после блок-пакета p, или – в случае конечного блок-пакета – Программного потока, время прибытия байта, содержащего последний бит кода MPEG_program_end_code.

Размер буфера декодера

В случае потока CSPS ограничивается максимальный размер каждого входного буфера в декодере конечной системы. Ограничения для элементарных видеопотоков отличаются от ограничений для элементарных аудиопотоков.

В случае передачи в потоке CSPS элементарного видеопотока, соответствующего Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 11172-2, применяется следующее правило:

Буфер BS_n имеет размер, который равен сумме размеров буфера Video Buffer Verifier (VBV), определенного для потока, соответствующего Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 11172-2, соответственно, и дополнительного объема буферизации BS_{add}. BS_{add} определяется как:

$$BS_{add} \le MAX\,[6 \times 1024,\ R_{vmax} \times 0{,}001] \text{ байтов,}$$

где R_{vmax} – максимальная двоичная скорость элементарного видеопотока, соответствующего Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или ИСО/МЭК 11172-2.

В случае передачи в потоке CSPS элементарного видеопотока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, применяется следующее правило:

Буфер BS_n имеет размер, равный сумме значений cpb_size и дополнительного объема буферизации BS_{add}. BS_{add} определяется как:

$$BS_{add} \le MAX\,[6 \times 1024,\ R_{vmax} \times 0{,}001] \text{ байтов,}$$

где R_{vmax} – максимальная двоичная скорость видеопотока AVC и

где cpb_size представляет собой CpbSize[cpt_cnt_minus1] размер CPB для формата потока байтов, передаваемого в параметре NAL hrd_parameters() видеопотока AVC. Если параметр NAL hrd_parameters() в видеопотоке AVC отсутствует, то поле cpb_size должно содержать размер, определенный как 1200 × MaxCPB в Приложении A Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 для используемого уровня.

В случае передачи в потоке CSPS элементарного аудиопотока, применяется следующее правило:

$$BS_n \le 4096 \text{ байтов}$$

В случае передачи в потоке CSPS элементарного аудиопотока, соответствующего стандарту ИСО/МЭК 13818-7 ADTS, применяется следующее правило для обеспечения передачи 8 каналов:

$$BS_n \le 8976 \text{ байтов}$$

2.7.10 Транспортный поток

Синхронизация частоты дискретизации в Транспортных потоках

В Транспортном потоке должна быть определена постоянная разумная взаимосвязь между частотой дискретизации аудиосигнала и частотой системных часов в декодере конечной системы, и, аналогично, определена постоянная разумная взаимосвязь между частотой видеокадров и частотой системных часов. Частота системных часов (system_clock_frequency) определяется в § 2.4.2. Частота видеокадров определена в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 или в ИСО/МЭК 11172-2. Частота дискретизации аудиосигнала определена в ИСО/МЭК 13818-3 или в ИСО/МЭК 11172-3. Для всех модулей представления элементарных аудиопотоков в составе Транспортного потока отношение system_clock_frequency к фактической частоте дискретизации аудиосигнала (SCASR) постоянно и равно значению, приведенному в следующей таблице для номинальной частоты дискретизации, указанной в аудиопотоке.

$$SCASR = \frac{system_clock_frequency}{audio_sample_rate_in_the_T-STD} \tag{2-30}$$

Нотация $\frac{X}{Y}$ означает деление вещественных чисел.

Номинальная частота дискретизации аудиосигнала (кГц)	16	32	22,05	44,1	24	48
SCASR	27 000 000 / 16 000	27 000 000 / 32 000	27 000 000 / 22 050	27 000 000 / 44 100	27 000 000 / 24 000	27 000 000 / 48 000

Для всех модулей представления всех видеопотоков в Транспортном потоке, соответствующем ИСО/МЭК 11172-2 и Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, отношение частоты системных часов (system_clock_frequency) к фактической частоте видеокадров (SCFR) постоянно и равно значению, приведенному в следующей таблице для номинальной частоты кадров, указанной в видеопотоке.

$$SCFR = \frac{system_clock_frequency}{frame_rate_in_the_T-STD} \tag{2-31}$$

Номинальная частота кадров (Гц)	23,976	24	25	29,97	30	50	59,94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

Значения SCFR являются точными. Реальная частота кадров слегка отличается от номинальной частоты в тех случаях, когда номинальная частота равна 23,976, 29,97 или 59,94 кадров в секунду.

Для видеопотоков, соответствующих ИСО/МЭК 14496-2, передаваемых в составе транспортного потока, временная база видеопотоков, соответствующих ИСО/МЭК 14496-2, определенная в параметре vop_time_increment_resolution, должна быть синхронизирована с STC и быть в точности равной N-кратному значению частоты system_clock_frequency, поделенному на K, где N и K – целые числа, имеющие фиксированные значения для каждой последовательности визуальных объектов, причем K больше или равно N.

Для видеопотоков, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, временная база видеопотока должна быть синхронизирована с частотой системных часов. Частота временной базы AVC определяется параметром time_scale потока AVC и должна быть в точности равна N-кратному значению частоты system_clock_frequency, поделенному на K, где N и K – целые числа, имеющие фиксированные значения для каждой видеопоследовательности AVC, причем K больше или равно N. Например, если параметр time_scale установлен в значение 90 000, то частота временной базы AVC точно равна значению system_clock_frequency, поделенному на 300.

2.8 Совместимость с ИСО/МЭК 11172

Программный поток, соответствующий настоящей Рекомендации | Международному стандарту, определяется так, чтобы он был совместим вперед со стандартом ИСО/МЭК 11172-1. Декодеры Программного потока, определенные в настоящей Рекомендации | Международном стандарте, также должны поддерживать декодирование стандарта ИСО/МЭК 11172-1.

2.9 Регистрация идентификаторов авторских прав

2.9.1 Общие положения

В Частях 1, 2 и 3 стандарта ИСО/МЭК 13818 описаны правила управления авторскими правами на аудиовизуальные произведения. В Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 это осуществляется при помощи дескриптора авторских прав, тогда как в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 и ИСО/МЭК 13818-3 описываются поля для идентификации владельцев авторских прав посредством синтаксических полей в синтаксисе элементарного потока. В настоящей Рекомендации | Международном стандарте представлен метод получения и регистрации идентификаторов авторских прав, в соответствии с Рекомендацией МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

В Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 определен уникальный 32-битовый идентификатор авторских прав copyright_identifier, который является идентификатором типа произведения (таким, как ISBN, ISSN, ISRC и т. п.), передаваемым в дескрипторе авторских прав. Идентификатор copyright_identifier позволяет осуществить идентификацию большого числа Уполномоченных органов по регистрации авторских прав. Каждый Уполномоченный орган по регистрации авторских прав может определить синтаксис и семантику для идентификации аудиовизуальных произведений или других произведений, требующих защиты авторских прав, за счет соответствующего использования поля переменной длины additional_copyright_info, которое содержит номер защиты авторских прав.

В последующем подразделе и в Приложениях L, M и N описаны преимущества и обязанности всех участников регистрации идентификатора copyright_identifier.

2.9.2 Формирование органа по регистрации (RA)

Объединенная группа ИСО/МЭК JTC 1 должна потребовать назначения международной организации, которая будет работать в качестве органа по регистрации идентификаторов авторских прав **copyright_identifier,** определенных в § 2.6.24. Выбранная организация должна действовать как орган по регистрации. Так называемый орган по регистрации должен выполнять свои обязанности в соответствии с указаниями, приведенными в Приложении H/JTC 1. Зарегистрированный идентификатор copyright_identifier далее называется зарегистрированным идентификатором (RID).

После того, как орган по регистрации выбран, объединенная группа JTC 1 должна потребовать создания Группы управления регистрацией (RMG), которая будет рассматривать заявления, поданные организациями, чьи запросы о назначении RID, которые должны были использоваться в соответствии с Рекомендацией МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, были отклонены органом по регистрации.

Приложения L, M и N содержат информацию о процедурах регистрации уникального идентификатора авторских прав.

2.10 Регистрация формата конфиденциальных данных

Дескриптор регистрации, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, приводится в данном тексте, для того чтобы дать возможность пользователям настоящей Спецификации, однозначно передавать данные, когда формат этих данных не признается настоящей Спецификацией. Это положение позволит настоящей Спецификации передавать все типы данных, обеспечивая при этом однозначную идентификацию характеристик вложенной в них конфиденциальной информации.

2.10.1 Общие положения

В последующем подразделе и в Приложениях O и P описаны преимущества и обязанности всех участников процесса регистрации формата конфиденциальных данных.

2.10.2 Формирование органа по регистрации (RA)

Объединенная группа ИСО/МЭК JTC 1/SC 29 должна потребовать назначения одной из организаций – членов ИСО или Национальных комитетов МЭК, которая будет работать в качестве органа по регистрации идентификаторов формата **format_identifier,** определенных в § 2.6.8 и § 2.6.9. Выбранная организация должна действовать как орган по регистрации. Так называемый орган по регистрации должен выполнять свои обязанности в соответствии с указаниями в Приложении H/JTC 1. Зарегистрированный идентификатор формата конфиденциальной информации format_identifier далее называется зарегистрированным идентификатором (RID).

После того, как орган по регистрации выбран, объединенная группа JTC 1 должна потребовать создания Группы управления регистрацией (RMG), которая будет рассматривать заявления, поданные организациями, чьи запросы о назначении RID, которые должны были использоваться в соответствии с настоящей Спецификацией, были отклонены органом по регистрации.

Приложения O и P содержат информацию о процедурах регистрации уникального идентификатора формата.

2.11 Передача данных в соответствии с ИСО/МЭК 14496

2.11.1 Введение

Поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, может содержать отдельные элементарные потоки, соответствующие ИСО/МЭК 14496-2 и 14496-3, а также аудиовизуальные сцены, соответствующие ИСО/МЭК 14496-1, и связанные с ними потоки. Как правило, потоки, соответствующие ИСО/МЭК 14496, являются элементами программы, соответствующей Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, как определено таблицей PMT в Транспортном потоке и картой PSM в Программном потоке.

Для передачи данных, соответствующих ИСО/МЭК 14496, в Транспортных и Программных потоках делается различие между отдельными элементарными потоками и аудиовизуальными сценами, соответствующими ИСО/МЭК 14496-1, и связанными с ними потоками. Для передачи отдельных потоков ИСО/МЭК 14496-2 и элементарных потоков 14496-3 используются только системные инструменты, описанные в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, как определено в § 2.11.2. Для передачи аудиовизуальной сцены ИСО/МЭК 14496-1 и соответствующих элементарных потоков ИСО/МЭК 14496, содержащихся в потоках, сформированных из пакетов SL, соответствующих ИСО/МЭК 14496-1, или в потоках типа FlexMux, используются инструменты, описанные как в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, так и в ИСО/МЭК 14496-1, как определено в § 2.11.3.

Передача видеосигналов, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, в потоках, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, определена в § 2.14.

2.11.2 Передача отдельных элементарных потоков, соответствующих ИСО/МЭК 14496-2 и 14496-3, в пакетах PES

2.11.2.1 Введение

Отдельные элементарные потоки, соответствующие ИСО/МЭК 14496-2 и 14496-3, могут передаваться в пакетах PES в виде байтов PES_packet_data_byte. При формировании потока пакетов PES на синхронизацию данных никаких специальных ограничений не налагается. Для обеспечения синхронизации, в заголовке пакета PES, который содержит данные элементарных потоков, соответствующих ИСО/МЭК 14496, кодируются метки PTS и, если применимо, метки DTS; для кодирования меток PTS и DTS применяются те же ограничения, что и для элементарных потоков, соответствующих ИСО/МЭК 13818. В таблице 2-93 приведен обзор способов передачи отдельных потоков, соответствующих ИСО/МЭК 14496, в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

Таблица 2-93 – Передача отдельных потоков, соответствующих ИСО/МЭК 14496 в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1

ИСО/МЭК 14496-2 видео	Передача в пакетах PES	Stream_type = 0x10	Stream_id = '1110 xxxx'
ИСО/МЭК 14496-3 аудио	Передача в пакетах PES	Stream_type = 0x11	Stream_id = '110x xxxx'

Если в заголовке пакета PES присутствует метка PTS или DTS, она должна указывать визуальный объект, который следует либо после первого стартового кода VOP, либо после первого стартового кода объекта с неподвижной текстурой, который начинается в пакете PES. Каждый видеопоток, соответствующий ИСО/МЭК 14496-2, передаваемый в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, должен содержать информацию, требуемую для декодирования видеопотока, соответствующего ИСО/МЭК 14496-2; следовательно, этот поток должен содержать заголовки последовательности визуальных объектов, заголовки визуальных объектов и заголовки уровней визуальных объектов.

В случае передачи элементарного потока, соответствующего ИСО/МЭК 14496-3, до формирования пакетов PES, данные элементарных потоков должны быть сначала инкапсулированы в объекты транспортного синтаксиса LATM/LOAS AudioSyncStream(), определенные в ИСО/МЭК 14496-3. Если в заголовке пакета PES присутствует метка PTS, она должна указывать первый аудиокадр, следующий после первого синхрослова, которое начинается в полезной нагрузке пакета PES.

Передача в пакетах PES элементарных потоков, соответствующих ИСО/МЭК 14496-2 и ИСО/МЭК 14496-3, должна быть определена значениями stream_id и stream_type, указывающими на использование видеопотоков, соответствующих ИСО/МЭК 14496-2, или аудиопотоков, соответствующих 14496-3. Кроме того, о такой передаче должен сообщать, соответственно, дескриптор MPEG-4_video или дескриптор MPEG-4_audio. Эти дескрипторы для соответствующего входящего элементарного потока должны передаваться в цепочке дескрипторов в таблице преобразования программы (в случае транспортного потока) или в карте программного потока, если она имеется (в случае программного потока). Рекомендация МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 не определяет воспроизведения элементарных потоков ИСО/МЭК 14496-2 и ИСО/МЭК 14496-3 в контексте программы.

2.11.2.2 Расширения декодера STD для отдельных элементарных потоков, соответствующих ИСО/МЭК 14496

Модель декодера T-STD содержит транспортный буфер TB_n и буфер мультиплексирования B_n, расположенные до декодирования каждого отдельного элементарного потока "n", соответствующего ИСО/МЭК 14496. Отметим, что в декодере T-STD для видеопотока, соответствующего ИСО/МЭК 14496-2, применяется отдельный буфер мультиплексирования B_n, как показано на рисунке 2-4, вместо подхода с двумя буферами MB_n и EB_n, используемого в T-STD для видеопотока, соответствующего ИСО/МЭК 13818-2. Для буферов TB_n и B_n и скорости передачи Rx_n между буферами TB_n и B_n применяются следующие ограничения.

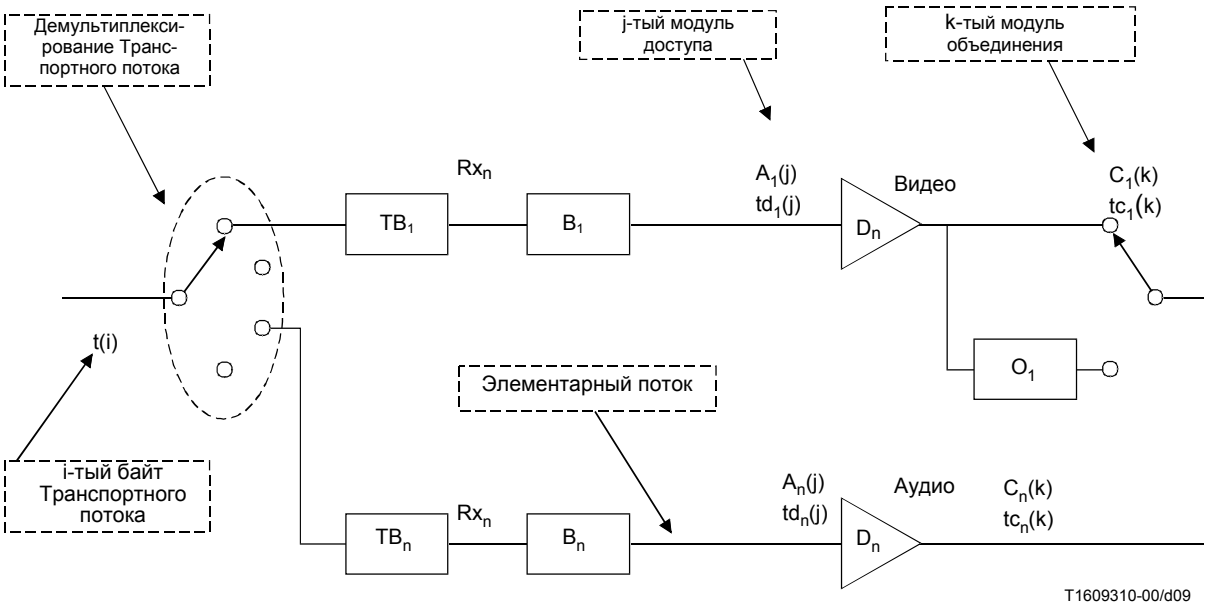


Рисунок 2-4 – Расширения модели декодера STD для отдельных элементарных потоков, соответствующих ИСО/МЭК 14496

В случае передачи потока, соответствующего ИСО/МЭК 14496-2:

Размер BS_n буфера B_n:

$$BS_n = BS_{mux} + BS_{oh} + VBV_{max}[profile,level],$$

где:

BS_{oh}, буферизация избыточных данных пакета, определяется следующим образом:

BS_{oh} = (1/750) секунд × max{R_{max}[profile,level], 2 000 000 бит/с}

и:

BS_{mux}, дополнительная буферизация мультиплексирования, определяется следующим образом:

BS_{mux} = 0,004 секунд × max{R_{max}[profile,level], 2 000 000 бит/с}

Скорость Rx_n:

$$Rx_n = 1.2 \times R_{max}[profile,level],$$

где:

VBV_{max}[profile,level] и R_{max}[profile,level] определяются в ИСО/МЭК 14496-2 для каждого профиля и уровня. Для профилей и уровней, для которых значение VBV_{max} не определено, размер B_n и скорость Rx_n определяются пользователем.

В случае передачи потока, соответствующего ИСО/МЭК 14496-3:

Размер BS_n буфера B_n для аудиопотока AAC, соответствующего ИСО/МЭК 14496-3.

$$\text{в ином случае } BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \text{ байтов}$$

В таком случае размер буфера для модуля доступа декодирования BS_{dec} и размер буфера избыточных данных пакета PES BS_{oh} ограничиваются значением:

$$BS_{dec} + BS_{oh} \leq 2848 \text{ байтов}$$

Часть (736 байтов) буфера размером 3584 байтов предназначена для буферизации, которая позволяет выполнять мультиплексирование. Оставшиеся 2848 байтов делятся между буферизацией модуля доступа BS_{dec}, BS_{oh} и дополнительной буферизацией мультиплексирования.

Скорость Rx_n для аудиопотока AAC, соответствующего ИСО/МЭК 14496-3, такая же, как определена § 2.4.2.3 для аудиопотока ADTS, соответствующего ИСО/МЭК 13818-7:

$$\text{в ином случае } Rx_n = 2\,000\,000 \text{ бит/с}$$

Модель декодера P-STD содержит буфер мультиплексирования B_n до декодирования каждого отдельного элементарного потока "n", соответствующего ИСО/МЭК 14496. Размер BS_n буфера B_n в декодере P-STD определяется полем P-STD_buffer_size в заголовке пакета PES.

2.11.3 Передача аудиовизуальных сцен, соответствующих ИСО/МЭК 14496-1, и связанных с ними потоков, соответствующих ИСО/МЭК 14496

2.11.3.1 Введение

В настоящем подразделе описывается инкапсуляция и сигнализация для случая, когда аудиовизуальная сцена, представленная в виде данных, соответствующих ИСО/МЭК 14496, передается в Программном или Транспортном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Содержание, соответствующее ИСО/МЭК 14496, состоит из исходного дескриптора объекта и переменного числа таких потоков, как потоки дескрипторов объекта, потоки описания сцены (в которых передаются либо команды BIFS, либо модули доступа BIFS-Anim), потоки IPMP, потоки OCI и аудиовизуальные потоки. Каждый из потоков, соответствующих ИСО/МЭК 14496, должен содержаться в потоке, сформированном из пакетов SL, и они могут быть дополнительно мультиплексированы в поток FlexMux, оба эти потока определены в ИСО/МЭК 14496-1. Для передачи Программного или Транспортного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, эти потоки, сформированные из пакетов SL, и потоки FlexMux должны содержать закодированные поля Эталонного времени объекта – Object Clock Reference (OCR) и Эталонного времени FlexMux – FlexMux Clock Reference (FCR), определенные в § 2.11.3.4 и § 2.11.3.5, соответственно. Потоки, сформированные из пакетов SL, или потоки FlexMux затем инкапсулируются либо в пакеты PES, либо в секции ISO_IEC_14496_sections, после чего выполняется деление на пакеты и мультиплексирование Транспортного потока или мультиплексирование Программного потока. Секции ISO_IEC_14496_sections формируются на длинноформатных секций H.222.0 | ИСО/МЭК 13818-1.

2.11.3.2 Выравнивание значений ES_ID

Сцена, соответствующая ИСО/МЭК 14496-1, передаваемая в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, может связывать множество потоков стандарта ИСО/МЭК 14496, ИСО/МЭК 13818 и других потоков за счет использования параметра ES_ID. Сцена и связанные с ней потоки могут передаваться в одном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, но сцена может указывать потоки, передаваемые где-либо еще, например, в сети IP. В настоящей Спецификации не определяется, как идентифицировать эти другие способы передачи.

В стандарте ИСО/МЭК 14496-1 определяются правила обзора имен для идентификаторов. Эти правила позволяют использовать одно и то же значение ES_ID для двух различных потоков внутри содержания, соответствующего ИСО/МЭК 14496. Когда в программном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, передается одна или несколько сцен, значения ES_ID внутри программы не должны дублироваться, так, чтобы поток, сформированный из пакетов SL и соответствующий ИСО/МЭК 14496, или канал FlexMux, соответствующий ИСО/МЭК 14496-1. имели бы в данной программе уникальное значение ES_ID.

2.11.3.3 Синхронизация сцен, соответствующих ИСО/МЭК 14496, и связанных с ними потоков

При передаче в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, временная база объекта каждого потока, соответствующего ИСО/МЭК 14496, должна быть синхронизирована с временной базой потока STC, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, то есть:

$$\text{Если } X(t) = f_{stc}(t)/f_{object}(t),$$

то значение X(t) должно быть постоянным в любой момент времени t,

где:

$f_{stc}(t)$ обозначает заданную частоту STC в момент времени t, т. е. 27 000 000 Гц

$f_{object}(t)$ обозначает частоту временной базы объекта в момент времени t

Временная база объекта потоков, соответствующих ИСО/МЭК 14496, передаваемых в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, передается следующим образом:

- Временная база объекта потока, сформированного из пакетов SL, передаваемых в пакетах PES без использования FlexMux, должна передаваться посредством кодированных значений Эталонного времени объекта (OCR) в заголовке пакета SL этого потока. См. § 2.11.3.4.
- Временная база объекта потоков, сформированных из пакетов SL, передаваемых в пакетах PES внутри потока FlexMux, должна передаваться посредством значений Эталонного времени FlexMux (FCR) в этом потоке FlexMux. См. § 2.11.3.5. Следовательно, все потоки, соответствующие ИСО/МЭК 14496, содержащиеся внутри одного потока FlexMux, используют совместно одну и ту же временную базу объекта.
- Временная база объекта потока, сформированного из пакетов SL, передаваемых в секциях, должна передаваться в Транспортном или Программном потоке посредством другого потока, соответствующего ИСО/МЭК 14496, как указано в поле OCR_ES_ID дескриптора ES для этого потока.

Приведенные далее ограничения должны применяться для кодирования значений OCR и FCR в потоках, сформированных из пакетов SL, и потоках FlexMux, передаваемых в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1:

- Значения OCR и FCR в каждом потоке, сформированном из пакетов SL, и каждом потоке FlexMux, связанные с одной и той же сценой, должны иметь одно и то же разрешение.
- Разрешение значений OCR и FCR для некоторой сцены (f_{cr}) должно быть меньше или равно 90 000 Гц.
- Отношение $(f_{stc}(t)/300)/f_{cr}$ должно быть целочисленным значением, большим или равным единице. Следовательно, разрешение синтаксических элементов OCR и FCR может принимать только такие значения, как 90 000 Гц, 45 000 Гц, 30 000 Гц, 22 500 Гц, 18 000 Гц и т. п.

В рамках вышеуказанных ограничений и ограничений стандарта ИСО/МЭК 14496-1, согласно которым разрешение f_{cr} должно быть целым числом циклов в секунду, может быть выбрана такая величина f_{cr}, которая приемлема для данной сцены.

Метки времени, соответствующие ИСО/МЭК 14496, закодированные в заголовке пакета SL, должны указывать моменты временной базы объекта потока, передаваемого в пакете SL. Разрешение каждой такой метки времени должно быть в 2^k раз меньше, чем разрешение элементов OCR или FCR, связанных с этим потоком, где k – положительное целое число, которое больше или равно нулю. Для того же самого времени цикла, поля длина метки времени (TimeStampLength) должна быть на k битов меньше, чем длина поля OCR или FCR (OCRLength и FCRLength, соответственно). Следовательно, для каждого потока должны применяться следующие условия кодирования меток времени:

- TimeStampResolution = (OCRResolution или FCRResolution, соответственно)/2^k, где k – положительное целое число, которое больше или равно нулю. В стандарте ИСО/МЭК 14496-1 требуется, чтобы значение TimeStampResolution представляло собой целое число циклов в секунду.
- TimeStampLength = (OCRLength или FCRLength, соответственно) – k.

Соотношение между значением STC и соответствующим значением временной базы объекта некоторого потока устанавливается путем связывания полей PTS в заголовках пакетов PES со значениями OCR или FCR в заголовках пакетов SL и пакетов потока FlexMux, соответственно, как определено в § 2.11.3.6 и § 2.11.3.7.

2.11.3.4 Синхронизация доставки потоков, сформированных из пакетов SL

Для передачи содержания, соответствующего ИСО/МЭК 14496, в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, используются сформированные из пакетов SL потоки, соответствующие ИСО/МЭК 14496-1. В каждом потоке, сформированном из пакетов SL, передаваемом в пакете PES без использования FlexMux, поле objectClockReference должно быть закодировано следующим образом:

1) Поле objectClockReference (OCR) должно присутствовать в заголовке первого пакета SL потока, сформированного из пакетов SL.

2) Поток, сформированный из пакетов SL, должен быть построен таким образом, чтобы интервал времени между байтами, содержащими последний бит последовательных полей OCR, был бы меньше или равен 0,7 с. Следовательно:

$$\left|t(i'') - t(i')\right| <= 0{,}7\,\text{с}$$

для всех i′ и i″, где i' и i″ – индексы байтов, содержащих последний бит последовательных полей OCR в потоке FlexMux.

Если в заголовке пакета SL закодировано поле objectClockReference, то должно быть закодировано также и поле instantBitrate.

2.11.3.5 Синхронизация доставки потоков FlexMux

Для передачи содержания, соответствующего ИСО/МЭК 14496, в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, после формирования потоков из пакетов SL может использоваться инструмент FlexMux, соответствующий ИСО/МЭК 14496-1. Полезная нагрузка пакетов FlexMux должна состоять из пакетов SL, как определено в ИСО/МЭК 14496-1. В каждом потоке FlexMux, передаваемом в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, поле fmxClockReference должно быть закодировано следующим образом:

1) Поле fmxClockReference (FCR) должно присутствовать в первом пакете FlexMux потока FlexMux.

2) Поток FlexMux должен быть построен таким образом, чтобы интервал времени между байтами, содержащими последний бит последовательных полей FCR, был бы меньше или равен 0,7 с. Следовательно:

$$\left|t(i'') - t(i')\right| <= 0{,}7\,\text{с}$$

для всех i′ и i″, где i' и i″ – индексы байтов, содержащих последний бит последовательных полей FCR в потоке FlexMux.

3) В потоках, сформированных из пакетов SL, передаваемых в потоке FlexMux, все метки времени, соответствующие ИСО/МЭК 14496, должны указывать моменты временной базы объекта, передаваемого в полях FCR потока FlexMux. Потоки, сформированные из пакетов SL, передаваемые в пакетах FlexMux, не должны содержать полей OCR. Если поля OCR присутствуют, они могут не приниматься во внимание.

2.11.3.6 Передача потоков, сформированных из пакетов SL, в пакетах PES

Отдельный поток, соответствующий ИСО/МЭК 14496-1, сформированный из пакетов SL, может быть преобразован в отдельный поток PES. Полезную нагрузку одного пакета PES должен составлять один и только один пакет SL из потока, сформированного из пакетов SL. Пакеты PES, в которых передается поток, сформированный из пакетов SL, должны быть идентифицированы в заголовке пакета PES идентификатором stream_id = 0xFA.

Когда в заголовке пакета SL закодировано поле OCR, в заголовке пакета PES, в котором передается этот заголовок пакета SL, должна быть закодирована метка PTS. Эта метка PTS должна быть закодирована при помощи 33-битового значения части метки STC (90 кГц), которая соответствует значению временной базы объекта в момент времени, указанный в поле OCR.

Идентификатор ES_ID, связанный с потоком, сформированным из пакетов SL, должен передаваться в дескрипторе SL, как определено в § 2.6.46.

2.11.3.7 Передача потоков FlexMux в пакетах PES

Пакеты PES, полезная нагрузка которых состоит из пакетов FlexMux, должны быть идентифицированы в заголовке пакета PES идентификатором stream_id = 0xFB. Полезную нагрузку одного пакета PES должно составлять целое число пакетов FlexMux, т. е. полезная нагрузка пакета PES, в котором передается поток FlexMux, должна начинаться с заголовка пакета FlexMux и должна завершаться последним байтом пакета FlexMux.

Если в одном из пакетов FlexMux, содержащихся в пакете PES закодировано поле fmxClockReference (FCR), то метка PTS должна быть закодирована в заголовке пакета PES, который содержит этот пакет FlexMux. Эта метка PTS должна быть закодирована при помощи 33-битового значения части метки STC (90 кГц), которая соответствует значению временной базы объекта потока FlexMux на момент времени, указанный в поле FCR. В том случае, когда в пакете PES имеется несколько пакетов FlexMux с кодированным полем FCR, метка PTS должна соответствовать времени, указанному в поле FCR первого такого пакета FlexMux, находящегося в полезной нагрузке пакета PES.

Идентификаторы ES_ID, связанные с каждым потоком, сформированным из пакетов SL, передаваемом в потоке FlexMux, должны передаваться в дескрипторе FMC, как определено в § 2.6.44.

2.11.3.8 Передача пакетов SL и пакетов FlexMux в секциях

Для транспортировки содержания, соответствующего ИСО/МЭК 14496, в секциях определяются секции ISO_IEC_14496_section. Секции ISO_IEC_14496_section должны использоваться только для потоков дескрипторов объектов, сформированных из пакетов SL, и потоков описания сцен. Отдельная секция ISO_IEC_14496_section должна содержать либо целый пакет SL потока, сформированного из пакетов SL, либо целое число пакетов FlexMux, в каждом из которых передается пакет SL одного и того же элементарного потока, соответствующего ИСО/МЭК 14496-1.

В таблице 2-94 показан синтаксис секций ISO_IEC_14496_section, определенный для передачи элементарных потоков, соответствующих ИСО/МЭК 14496-1, обозначенный идентификаторами table_id либо как данные потока дескрипторов объекта, либо как данные потока описания сцен. Данные потока дескрипторов объекта состоят из Таблицы дескрипторов объекта, которая содержит множество дескрипторов объекта. Таблица дескрипторов объекта может быть передана в нескольких секциях ISO_IEC_14496_section. Данные потока описания сцен состоят из Таблицы описания сцен, которая может содержать множество команд BIFS. Таблица описания сцен может быть передана в нескольких секциях ISO_IEC_14496_section. Для обработки полезной нагрузки не требуется, чтобы была принята полная таблица. Однако полезная нагрузка секций должна обрабатываться в правильном порядке, который указывается значением поля section_number в байтах заголовка секции ISO_IEC_14496_section.

Таблица 2-94 –Синтаксис секции для транспортировки потока, соответствующего ИСО/МЭК 14496

Синтаксис	Количество битов	Мнемоника
ISO_IEC_14496_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
зарезервировано	**2**	**bslbf**
ISO_IEC_14496_section_length	**12**	**uimsbf**
table_id_extension	**16**	**uimsbf**
зарезервировано	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
if (PMT_has_SL_descriptor(current_PID)) {		
SL_Packet()		
}		
else if (PMT_has_FMC_descriptor(current_PID)) {		
for (i = 1; i < N; i++)		
FlexMuxPacket()		
}		
else {		
for (i = 1; i < N; i++)		
зарезервировано	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		

table_id – Это 8-битовое поле должно быть установлено равным '0x04' или '0x05' в случае передачи секции ISO_IEC_14496_section. Значение '0x04' указывает секцию ISO_IEC_14496_scene_Description_section, в которой передается поток описания сцен, соответствующий ИСО/МЭК 14496-1. Значение '0x05' указывает секцию ISO_IEC_14496_object_descriptor_section, в которой передается поток дескрипторов объекта, соответствующий ИСО/МЭК 14496-1.

section_syntax_indicator – Это 1-битовое поле должно быть установлено равным '1'.

private_indicator – Это 1-битовое поле в настоящей Спецификации не определяется.

ISO_IEC_14496_section_length – Это 12-битовое поле должно определять количество оставшихся байтов в секции, которые следуют сразу после поля ISO_IEC_14496_section_length и до завершения секции ISO_IEC_14496_section. Значение этого поля не должно превышать 4093 (0xFFD).

table_id_extension – Это 16-битовое поле в настоящей Спецификации не определяется; его использование и его значение определяются пользователем.

version_number – Это 5-битовое поле должно представлять собой номер версии Таблицы дескрипторов объекта или Таблицы описания сцен, соответственно. Номер версии должен увеличиваться на 1 (по модулю 32) с добавлением в эту таблицу каждой новой версии. Управление версиями остается на усмотрение приложения.

current_next_indicator – Это 1-битовое поле должно быть установлено равным 1.

section_number – Это 8-битовое поле должно представлять собой номер секции ISO_IEC_14496_section. Поле section_number первой секции ISO_IEC_14496_section Таблицы дескрипторов объекта или Таблицы описания сцен должно иметь значение, равное 0x00. Значение номера section_number должно увеличиваться на 1 с добавлением в эту таблицу каждой дополнительной секции.

last_section_number – Это 8-битовое поле должно определять номер последней секции Таблицы дескрипторов объекта или Таблицы описания сцен, частью которой является эта секция.

PMT_has_SL_descriptor(current_PID) – Псевдо-функция, которая должна иметь значение "true", если в цепочке дескриптора Таблицы преобразования программы для программного элемента, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, в котором передается эта секция ISO_IEC_14496_section, содержится дескриптор SL.

SL_Packet() – Пакет синхроуровня, определенный в § 10.2.2 стандарта ИСО/МЭК 14496-1.

PMT_has_FMC_descriptor(current_PID) – Псевдо-функция, которая должна иметь значение "true", если в цепочке дескриптора Таблицы преобразования программы для программного элемента, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, в котором передается эта секция ISO_IEC_14496_section, содержится дескриптор FMC.

FlexMuxPacket() – Пакет FlexMux, определенный в § 11.2.4 стандарта ИСО/МЭК 14496-1.

CRC_32 – Это 32-битовое поле должно содержать значение CRC, которое после обработки полной секции ISO_IEC_14496_section дает ноль на выходе регистров декодера, определенного в Приложении A.

2.11.3.9 Расширения декодера T-STD

2.11.3.9.1 Модель T-STD для содержания, соответствующего ИСО/МЭК 14496

На рисунке 2-5 показаны расширения Транспортного декодера конечной системы для доставки программного элемента, соответствующего ИСО/МЭК 14496, инкапсулированного в Транспортные потоки, соответствующие Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

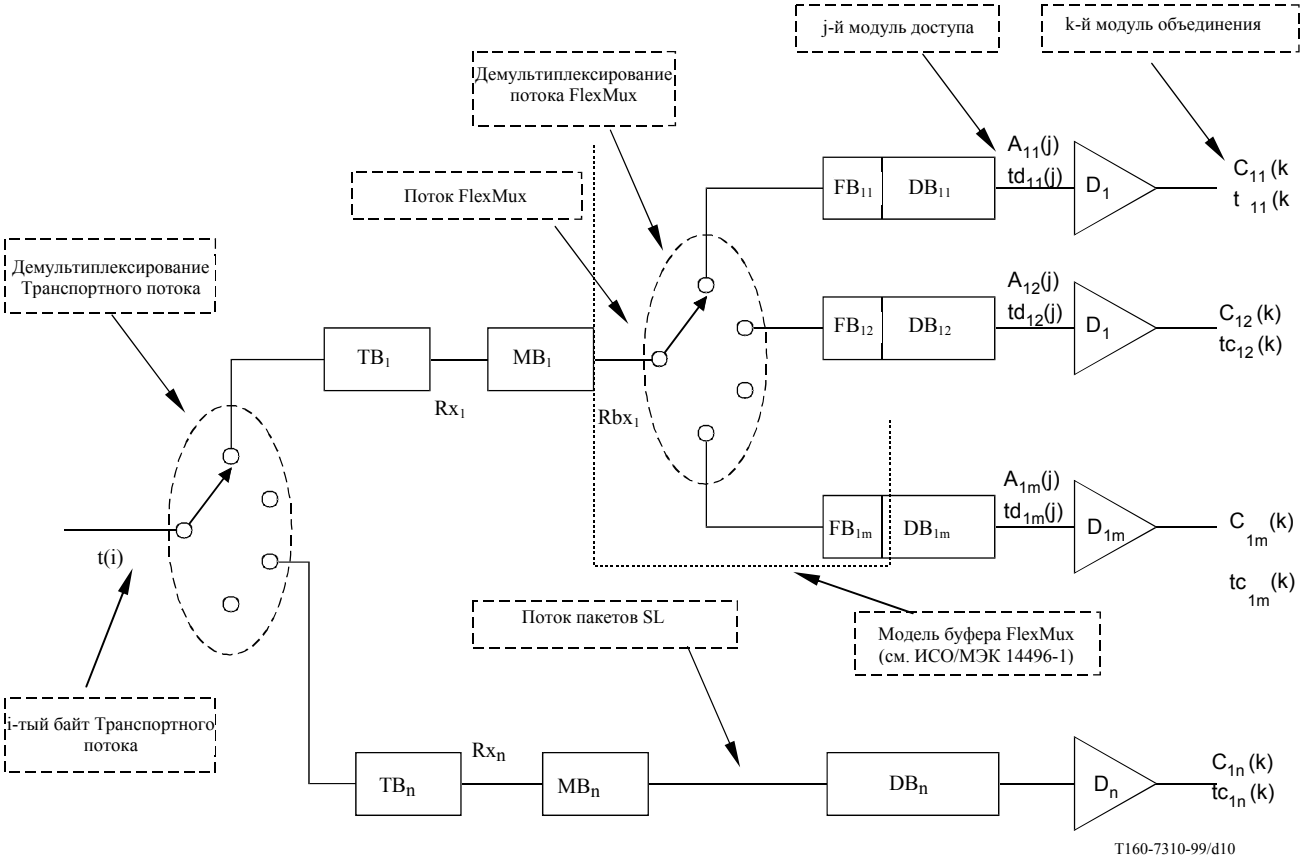


Рисунок 2-5 – Модель декодера T-STD для содержания, соответствующего ИСО/МЭК 14496

На рисунке 2-5 и в его описании использованы следующие обозначения:

TB_n транспортный буфер.

MB_n буфер мультиплексирования для потока "n" FlexMux или потока "n", сформированного из пакетов SL.

FB_{np} буфер FlexMux для элементарного потока в канале FlexMux "p" потока "n" FlexMux.

DB_{np} буфер декодера для элементарного потока в канале FlexMux "p" потока "n" FlexMux.

DB_n буфер декодера для элементарного потока "n".

D_{np} декодер для элементарного потока в канале FlexMux "p" потока "n" FlexMux.

D_n декодер для элементарного потока n.

Rx_n скорость, с которой данные удаляются из TB_n.

Rbx_n скорость, с которой данные удаляются из MB_n.

$A_{np}(j)$ j-тый модуль доступа в элементарном потоке в канале FlexMux "p" потока "n" FlexMux. $A_{np}(j)$ нумеруется в порядке декодирования.

$A_n(j)$ j-тый модуль доступа в элементарном потоке "n". $A_n(j)$ нумеруется в порядке декодирования.

$Td_{np}(j)$ измеренное в секундах время декодирования в декодере конечной системы для j-того модуля доступа в элементарном потоке в канале FlexMux "p" потока "n" FlexMux.

$Td_n(j)$ измеренное в секундах время декодирования в декодере конечной системы для j-того модуля доступа в элементарном потоке "n".

$C_{np}(k)$ j-тый модуль объединения в элементарном потоке в канале FlexMux "p" потока "n" FlexMux. $C_{np}(k)$ получается в результате декодирования $A_{np}(j)$. $C_{np}(k)$ нумеруется в порядке объединения.

$C_n(k)$ j-тый модуль объединения в элементарном потоке "n". $C_n(k)$ получается в результате декодирования $A_n(j)$. $C_n(k)$ нумеруется в порядке объединения.

$tc_{np}(k)$ измеренное в секундах время объединения в декодере конечной системы для j-того модуля объединения в элементарном потоке в канале FlexMux "p" потока "n" FlexMux.

$tc_n(k)$ измеренное в секундах время объединения в декодере конечной системы для j-того модуля объединения в элементарном потоке "n".

t(i) указывает время в секундах, когда i-тый байт Транспортного потока поступает в декодер конечной системы.

2.11.3.9.2 Обработка потоков FlexMux

Полные пакеты Транспортного потока, содержащие данные из потока FlexMux "n", передаются в транспортный буфер для потока "n" FlexMux (TB_n). Размер буфера TB_n фиксирован и равен 512 байтов. Все байты, которые поступают в TB_n, удаляются из TB_n со скоростью Rx_n, определенной полем TB_leak_rate в дескрипторе MultiplexBuffer, связанном с потоком FlexMux "n". Когда в буфере TB_n нет данных, скорость Rx_n равна нулю. Дубликаты пакетов Транспортного потока в MB_n не доставляются.

В том случае, когда передача ведется в пакетах PES, заголовок пакета PES и байты полезной нагрузки доставляются в буфер MB_n; все остальные байты, покидающие буфер TB_n, в буфер MB_n не подаются, и могут использоваться для управления системой. В том случае, когда передача ведется в секциях ISO_IEC_14496_section, заголовок секции, байты полезной нагрузки и байты данных CRC-32 доставляются в буфер MB_n; все остальные байты в буфер MB_n не подаются, и могут использоваться для управления системой. В любом случае, размер буфера MB_n должен быть определен полем MB_buffer_size в дескрипторе MultiplexBuffer.

Все байты пакета потока FlexMux, находящиеся в буфере MB_n, доставляются в связанный с ними буфер FlexMux со скоростью, определенной полем fmxRate, закодированном в потоке FlexMux, и в соответствии с моделью буфера FlexMux, определенной в § 11.2.9 стандарта ИСО/МЭК 14496-1. В буфер FB_{np} поступают только байты полезной нагрузки пакета FlexMux, передаваемые в канале FlexMux "p" потока "n" FlexMux. Байты заголовка пакета FlexMux, передаваемые в канале FlexMux "p" потока "n" FlexMux, отбрасываются и могут использоваться для управления системой. Скорость, определенная полем fmxRate, должна быть применимой для всех пакетов FlexMux в потоке, которые следуют сразу после пакета канала эталонного времени FlexMux и до следующего пакета канала эталонного времени FlexMux. Когда в буфере MB_n нет данных из потока FlexMux, никаких данных из MB_n не удаляется. Байты из заголовка пакета PES или из заголовка секции ISO_IEC_14496_section, которые непосредственно предшествуют заголовку FlexMux, мгновенно удаляются и отбрасываются, и могут использоваться для управления системой. Байты из полей CRC-32 секции ISO_IEC_14496_section, которые следуют в полезной нагрузке секции сразу после последнего пакета потока FlexMux, мгновенно удаляются и отбрасываются, и могут использоваться для проверки целостности данных. Байты из канала эталонного времени FlexMux мгновенно удаляются и отбрасываются, и могут использоваться для синхронизации временной базы объекта, соответствующего ИСО/МЭК 14496, с STC. Когда в буфере MB_n нет байтов полезной нагрузки пакета PES или секции, соответственно, никакие данные из MB_n не удаляются. Все данные, которые поступают в MB_n, покидают его. Все байты полезной нагрузки пакета PES потока "n" сразу же после выхода из MB_n поступают в демультиплексор FlexMux.

2.11.3.9.3 Определение буфера FlexMux (FB_{np})

Для каждого канала "p" потока FlexMux "n", размер буфера FlexMux (FB_{np}) определяется с использованием дескриптора FmxBufferSize. Байты полезной нагрузки пакета FlexMux передаются из буфера FB_{np} в буфер декодера DB_{np} в соответствии с моделью буфера FlexMux, определенной в § 11.2.9 стандарта ИСО/МЭК 14496-1. В буфер DB_{np} поступают только байты полезной нагрузки пакета SL, передаваемые в канале FlexMux "p" потока "n" FlexMux. Байты заголовка пакета SL, передаваемые в канале FlexMux "p" потока "n" FlexMux, отбрасываются и могут использоваться для управления системой.

2.11.3.9.4 Обработка потоков, сформированных из пакетов SL

Полные пакеты Транспортного потока, содержащие данные из потока "n", сформированного из пакетов SL, передаются в транспортный буфер для потока "n", сформированного из пакетов SL (TB_n). Все байты, которые поступают в TB_n, удаляются из него со скоростью Rx_n, определенной полем TB_leak_rate в дескрипторе MultiplexBuffer. Когда в буфере TB_n нет данных, скорость Rx_n равна нулю. Дубликаты пакетов Транспортного потока в MB_n не доставляются.

В том случае, когда передача ведется в пакетах PES, заголовок пакета PES и байты полезной нагрузки доставляются в буфер MB_n; все остальные байты, покидающие буфер TB_n, в буфер MB_n не подаются, и могут использоваться для управления системой. В том случае, когда передача ведется в секциях ISO_IEC_14496_section, заголовок секции, байты полезной нагрузки и байты данных CRC-32 доставляются в буфер MB_n; все остальные байты в буфер MB_n не подаются и могут использоваться для управления системой. В любом случае, размер буфера MB_n определяется полем MB_buffer_size в дескрипторе MultiplexBuffer.

Все байты потока, сформированного из пакетов SL, находящиеся в буфере MB_n доставляются в буфер декодера DB_n со скоростью, определенной полем instantBitRate, закодированном в потоке, сформированном из пакетов SL, и в соответствии с моделью системного декодера, определенной в § 7.4 стандарта ИСО/МЭК 14496-1. Скорость, определенная полем instantBitRate, должна быть применимой для всех байтов данных в потоке, сформированном из пакетов SL, которые в заголовке пакета SL следуют сразу после поля instantBitRate и до следующего поля instantBitRate. Если в буфере MB_n нет байтов данных из потока, сформированного из пакетов SL, то никакие байты из MB_n не удаляются. Байты из заголовка пакета PES или из заголовка секции ISO_IEC_14496_section, которые непосредственно предшествуют заголовку пакета SL, мгновенно удаляются и отбрасываются, и могут использоваться для управления системой. Байты из полей CRC-32 секции ISO_IEC_14496_section, которые следуют в секции сразу же после последнего байта полезной нагрузки пакета SL, мгновенно удаляются и отбрасываются, и могут использоваться для проверки целостности данных. Когда в буфере MB_n нет байтов полезной нагрузки пакета PES или секции, соответственно, никакие данные из MB_n не удаляются. Все данные, которые поступают в MB_n, покидают его. Все байты полезной нагрузки пакета PES потока "n" сразу же после выхода из MB_n поступают в DB_n, за исключением заголовков пакетов SL. Байты из заголовков пакетов SL в буфер DB_n не подаются и могут использоваться для управления системой. Размер буфера декодера DB_n определяется полем bufferSizeDB дескриптора decoderConfigDescriptor, определенного в стандарте ИСО/МЭК 14496-1.

2.11.3.9.5 Управление буфером

Транспортные потоки должны быть сформированы так, чтобы выполнялись условия, определенные в настоящем подразделе.

Буфер CTB_n не должен переполняться и должен опустошаться, как минимум, раз в секунду Буфер. MB_n не должен переполняться. Буфер FB_{np} не должен переполняться. Буферы DB_{np} и DB_n не должны ни переполняться, ни быть недозаполненными. Состояние недозаполненности буфера DB_{np} возникает, когда во время декодирования, связанного с данным модулем доступа, в буфере DB_{np} отсутствует один или несколько байтов этого модуля доступа. Состояние недозаполненности буфера DB_n возникает, когда во время декодирования, связанного с данным модулем доступа, в DB_n отсутствует один или несколько байтов этого модуля доступа.

2.11.3.10 Передача в Транспортном потоке

2.11.3.10.1 Обзор

Транспортный поток может содержать одну или несколько программ, каждая из которых описывается Таблицей преобразования программы. В дополнение к типам потоков, которые уже определены для такой программы, может передаваться содержание, соответствующее ИСО/МЭК 14496. Элементы содержания, соответствующего ИСО/МЭК 14496, могут передаваться в одном или нескольких элементах программы, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, обозначенных в Транспортном потоке уникальным значением PID. В качестве особого случая, возможно, чтобы программа в Транспортном потоке содержала только элементы программы, соответствующие ИСО/МЭК 14496. Содержание, соответствующее ИСО/МЭК 14496, связанное с этой программой и передаваемое в Транспортном потоке, должно быть указано в Таблице преобразования программы для этой программы. Для определения сцены, соответствующей ИСО/МЭК 14496-1, должен использоваться исходный дескриптор объекта; использование этого дескриптора определяется в § 2.11.3.10.2.

О том, что в PID передается содержание, соответствующее ИСО/МЭК 14496, сообщается при помощи значения stream_type = 0x12 или 0x13 в Таблице преобразования программы, связанной с этим значением PID. Значение 0x12 означает передачу в пакетах PES. Поле stream_id в заголовке пакета PES сообщает о том, содержит ли пакет PES один пакет SL или несколько пакетов FlexMux. Значение stream_type = 0x13 в Таблице преобразования программы указывает, что программный элемент содержит поток дескрипторов объекта или поток команд BIFS, находящийся в секциях. В таком случае идентификатор table_id в заголовке секции указывает, передается ли поток дескрипторов объекта в секциях или в потоке команд BIFS. См. также таблицу 2-95. Секция содержит или один пакет SL, или несколько пакетов FlexMux, что указывается наличием либо дескриптора SL, либо дескриптора FMC, соответственно, в цепочке дескриптора Таблицы преобразования программы для программного элемента, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, в котором передаются эти секции. Когда передается содержание, соответствующее ИСО/МЭК 14496, дескриптор SL и дескриптор FMC должны определять идентификатор ES_ID для каждого инкапсулированного потока, соответствующего ИСО/МЭК 14496. Когда меняются назначенные значения идентификаторов ES_ID, Таблица преобразования программы (PMT) должна быть обновлена, и номер версии таблицы PMT (version_number) должен быть увеличен на 1 (по модулю 32). В Приложении R приведен пример процедуры доступа к содержанию для компонентов программы, соответствующих ИСО/МЭК 14496, в Транспортном потоке.

Таблица 2-95 – Определенные ИСО/МЭК варианты передачи сцены, соответствующей ИСО/МЭК 14496, и связанных с ней потоков, в потоке соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1

Потоки дескрипторов объектов, соответствующие ИСО/МЭК 14496-1	Инкапсуляция в пакеты SL	Передача в пакетах PES	Stream_type = 0x12	Stream_id = '1111 1010'
		Передача в секциях ISO_IEC_ 14496_section	Stream_type = 0x13	Table_id = 0x05
	Инкапсуляция в пакеты SL, за которой следует мультиплексирование в пакеты FlexMux	Передача в пакетах PES	Stream_type = 0x12	Stream_id = '1111 1011'
		Передача в секциях ISO_IEC_ 14496_section	Stream_type = 0x13	Table_id = 0x05
Потоки описания сцен, соответствующие ИСО/МЭК 14496-1	Инкапсуляция в пакеты SL	Передача в пакетах PES	Stream_type = 0x12	Stream_id = '1111 1010'
		Передача в секциях ISO_IEC_ 14496_section	Stream_type = 0x13	Table_id = 0x04
	Инкапсуляция в пакеты SL, за которой следует мультиплексирование в пакеты FlexMux	Передача в пакетах PES	Stream_type = 0x12	Stream_id = '1111 1011'
		Передача в секциях ISO_IEC_ 14496_section	Stream_type = 0x13	Table_id = 0x04
Все остальные потоки, соответствующие ИСО/МЭК 14496	Инкапсуляция в пакеты SL	Передача в пакетах PES	Stream_type = 0x12	Stream_id = '1111 1010'
	Инкапсуляция в пакеты SL, за которой следует мультиплексирование в пакеты FlexMux	Передача в пакетах PES	Stream_type = 0x12	Stream_id = '1111 1011'

2.11.3.10.2 Исходный дескриптор объекта

В случае передачи сцены в потоке, соответствующем ИСО/МЭК 14496-1, исходный дескриптор объекта, соответствующего ИСО/МЭК 14496-1, служит в качестве исходной точки доступа для всех связанных с ним потоков. Исходный дескриптор объекта должен передаваться в дескрипторе IOD, расположенном в цепочке дескриптора, которая следует сразу после поля program_info_length в Таблице преобразования программы для программы, с которой связана эта сцена. Она содержит дескрипторы ES_descriptor, обозначающие потоки описания сцен и дескрипторов объектов, которые являются частью этой программы. Она может также содержать дескрипторы ES_descriptor, обозначающие один или несколько связанных потоков IPMP или OCI. Идентификация потоков выполняется при помощи идентификаторов ES_ID, которые определены в разделе 8 стандарта ИСО/МЭК 14496-1.

2.11.3.11 Модель декодера P-STD для содержания, соответствующего ИСО/МЭК 14496

На рисунке 2-6 показана модель декодера STD для случая, когда в Программном потоке передается системная информация, соответствующая ИСО/МЭК 14496.

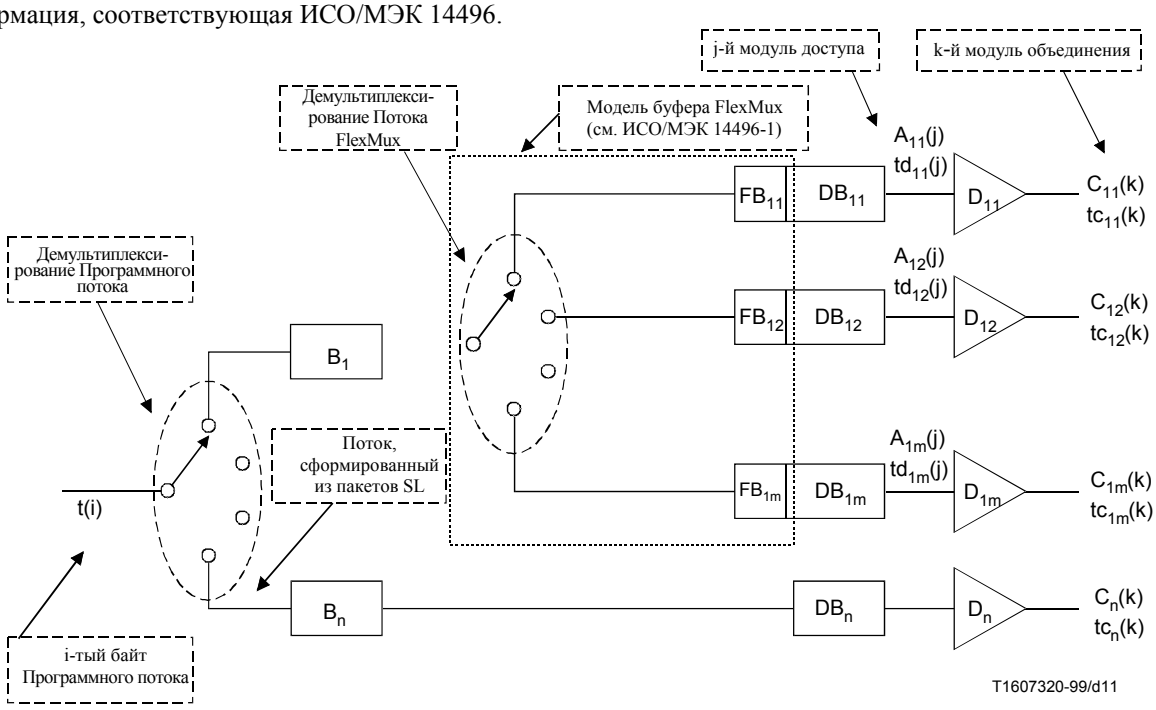


Рисунок 2-6 – Модель декодера P-STD для системного потока соответствующего ИСО/МЭК 14496

На рисунке 2-6 и в его описании использованы следующие обозначения:

B_n входной буфер для потока "n" FlexMux или для потока "n", сформированного из пакетов SL.

FB_{np} буфер FlexMux для элементарного потока в канале FlexMux "p" потока "n" FlexMux.

DB_{np} буфер декодера для элементарного потока в канале FlexMux "p" потока "n" FlexMux.

DB_n буфер декодера для элементарного потока "n".

D_{np} декодер для элементарного потока в канале FlexMux "p" потока "n" FlexMux.

D_n декодер для элементарного потока "n".

$A_{np}(j)$ j-тый модуль доступа в элементарном потоке в канале FlexMux "p" потока "n" FlexMux. $A_{np}(j)$ нумеруется в порядке декодирования.

$A_n(j)$ j-тый модуль доступа в элементарном потоке "n". $A_n(j)$ нумеруется в порядке декодирования.

$Td_{np}(j)$ измеренное в секундах время декодирования в декодере конечной системы для j-того модуля доступа в элементарном потоке в канале FlexMux "p" потока "n" FlexMux.

$Td_n(j)$ измеренное в секундах время декодирования в декодере конечной системы для j-того модуля доступа в элементарном потоке "n".

$C_{np}(k)$ j-тый модуль объединения в элементарном потоке в канале FlexMux "p" потока "n" FlexMux. $C_{np}(k)$ получается в результате декодирования $A_{np}(j)$. $C_{np}(k)$ нумеруется в порядке объединения.

$C_n(k)$ j-тый модуль объединения в элементарном потоке "n". $C_n(k)$ получается в результате декодирования $A_n(j)$. $C_n(k)$ нумеруется в порядке объединения.

$tc_{np}(k)$ измеренное в секундах время объединения в декодере конечной системы для j-того модуля объединения в элементарном потоке в канале FlexMux "p" потока "n" FlexMux.

$tc_n(k)$ измеренное в секундах время объединения в декодере конечной системы для j-того модуля объединения в элементарном потоке "n".

t(i) указывает время в секундах, когда i-тый байт Транспортного потока поступает в декодер конечной системы.

2.11.3.11.1 Обработка потоков FlexMux

На входе декодера STD каждый байт полезной нагрузки пакетов PES, в которых передается поток FlexMux "n", мгновенно передается в буфер B_n. i-тый байт поступает в B_n в момент времени t(i). Байты заголовка пакета PES не подаются в буфер B_n и могут использоваться для управления системой. Размер буфера B_n определен в поле P-STD_buffer_size в заголовке пакета PES, в котором передается поток "n".

Все байты пакета потока FlexMux, находящиеся в буфере B_n, доставляются в соответствующий им буфер FlexMux со скоростью, определенной полем fmxRate, закодированном в потоке FlexMux, и в соответствии с моделью буфера FlexMux, определенной в § 11.2.9 стандарта ИСО/МЭК 14496-1. В буфер FB_{np} поступают только байты полезной нагрузки пакета FlexMux, передаваемые в канале FlexMux "p" потока "n" FlexMux. Байты заголовка пакета FlexMux, передаваемые в канале FlexMux "p" потока "n" FlexMux, отбрасываются и могут использоваться для управления системой. Скорость, определенная полем fmxRate, должна быть применимой для всех пакетов FlexMux в потоке вплоть до следующего пакета канала эталонного времени FlexMux. Байты из канала эталонного времени FlexMux мгновенно удаляются и отбрасываются, и могут использоваться для синхронизации временной базы объекта, соответствующего ИСО/МЭК 14496, с STC. Когда в B_n нет данных полезной нагрузки пакета PES, из B_n не удаляются никакие данные. Все данные, которые поступают в B_n, покидают его. Все байты полезной нагрузки пакета PES потока "n" поступают в демультиплексор FlexMux сразу же после выхода из B_n.

2.11.3.11.2 Определение буфера FB_{np}

Если в Программном потоке присутствует карта Программного потока, то размер буфера FlexMux FB_{np} для каждого канала "p" потока FlexMux "n" определяется с использованием дескриптора FmxBufferSize. Байты полезной нагрузки пакета FlexMux передаются из буфера FB_{np} в буфер декодера DB_{np} в соответствии с моделью буфера FlexMux, определенной в § 11.2.9 стандарта ИСО/МЭК 14496-1. В буфер DB_{np} поступают только байты полезной нагрузки пакета SL, передаваемые в канале FlexMux "p" потока "n" FlexMux. Байты заголовка пакета SL, передаваемые в канале FlexMux "p" потока "n" FlexMux, отбрасываются и могут использоваться для управления системой.

2.11.3.11.3 Обработка потоков, сформированных из пакетов SL

На входе декодера STD каждый байт полезной нагрузки пакетов PES, в которых передается поток "n", сформированный из пакетов SL, мгновенно передается в буфер B_n. i-тый байт поступает в B_n в момент времени t(i). Байты заголовка пакета PES не подаются в буфер B_n и могут использоваться для управления системой. Размер буфера B_n определен в поле P-STD_buffer_size в заголовке пакета PES, в котором передается поток "n". Байты потока, сформированного из пакетов SL, в буфере B_n доставляются на буфер декодера DB_n со скоростью,

определенной полем instantBitRate, закодированном в потоке, сформированном из пакетов SL, и в соответствии с моделью системного декодера, определенной в § 7.4 стандарта ИСО/МЭК 14496-1. Скорость, определенная полем instantBitRate, должна быть применимой для всех байтов данных в потоке, сформированном из пакетов SL, вплоть до следующего поля instantBitRate. Когда в B_n нет данных полезной нагрузки пакета PES, из B_n не удаляются никакие данные. Все данные, которые поступают в B_n, покидают его. Все байты потока "n" поступают в буфер DB_n сразу же после выхода из B_n, за исключением заголовков пакетов SL. Байты заголовков пакетов SL не подаются в буфер DB_n и могут использоваться для управления системой. Размер буфера декодера DB_n определяется полем bufferSizeDB дескриптора DecoderConfigDescriptor, определенного в ИСО/МЭК 14496-1.

2.11.3.11.4 Управление буфером

Программные потоки должны быть сформированы так, чтобы буфер B_n не переполнялся. Буфер FB_{np} не должен переполняться. Буферы DB_{np} и DB_n не должны ни переполняться, ни быть недозаполненными. Состояние недозаполненности буфера DB_{np} возникает, когда во время декодирования, связанного с данным модулем доступа, в DB_{np} отсутствует один или несколько байтов этого модуля доступа. Состояние недозаполненности буфера DB_n возникает, когда во время декодирования, связанного с данным модулем доступа, в DB_n отсутствует один или несколько байтов этого модуля доступа.

2.11.3.12 Передача в Программном потоке

2.11.3.12.1 Обзор

Программный поток содержит только одну программу. В дополнение к типам потоков, которые уже определены для такой программы, могут передаваться данные, соответствующие ИСО/МЭК 14496. В качестве особого случая, допускается также, чтобы в Программном потоке передавались только данные, соответствующие ИСО/МЭК 14496. Если присутствует карта Программного потока, то передаваемое в Программном потоке содержание, соответствующее ИСО/МЭК 14496, должно быть указано следующим образом. Передача в пакетах SL и FlexMux сцен, соответствующих ИСО/МЭК 14496-1, и связанных с ними потоков ИСО/МЭК 14496 обозначается соответствующим значением идентификатора stream_id и исходным дескриптором объекта; использование этого дескриптора определяется в § 2.11.3.12.2. Для каждого передаваемого потока, соответствующего ИСО/МЭК 14496, идентификатор ES_ID должен определяться дескриптором SL и дескриптором FMC. В момент изменения назначенных значений ES_ID, карта Программного потока, если она присутствует, должна быть обновлена, и поле program_stream_map_version должно увеличиваться на 1 (по модулю 32). Отметим, что содержание, соответствующее ИСО/МЭК 14496, может быть также обозначено в Программном потоке при помощи конфиденциальных средств.

В Приложении R приведен пример процедуры доступа к содержанию для компонентов программы, соответствующих ИСО/МЭК 14496, передаваемых в Транспортном потоке.

2.11.3.12.2 Исходный дескриптор объекта

В случае передачи сцены в потоке, соответствующем ИСО/МЭК 14496-1, исходный дескриптор объекта, соответствующего ИСО/МЭК 14496-1, служит в качестве исходной точки доступа для всех связанных с ним потоков. Если в Программном потоке присутствует карта Программного потока, то исходный дескриптор объекта должен передаваться в дескрипторе IOD, расположенном в цепочке дескриптора, которая следует сразу после поля program_stream_info_length. Она содержит дескрипторы ES_descriptor, обозначающие потоки описания сцен и дескрипторов объектов, которые являются частью этой программы. Она может также содержать дескрипторы ES_descriptor, обозначающие один или несколько связанных потоков IPMP или OCI. Идентификация потоков выполняется при помощи идентификаторов ES_ID, которые определены в разделе 8 стандарта ИСО/МЭК 14496-1. Кроме того, исходный дескриптор объекта может передаваться в Программном потоке при помощи конфиденциальных средств.

2.12 Передача метаданных

2.12.1 Введение

Поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, может содержать метаданные. Формат метаданных может быть определен ИСО или любой другой уполномоченной организацией. В настоящем подразделе определяется, как передавать метаданные; определяются транспортные механизмы, а также сигнализация, связанная с метаданными, применяемая модель синхронизации метаданных и расширения модели декодера STD для декодирования метаданных.

Услуга метаданных определяется как когерентный набор метаданных одного формата, доставляемых на приемник с определенной целью. Услуги метаданных содержатся в потоках метаданных; в каждом потоке метаданных передается одна или несколько услуг метаданных. В настоящей Спецификации принимается понятие модулей доступа метаданных в рамках услуги метаданных. Определение модуля доступа метаданных зависит от формата метаданных, но предполагается, что каждая услуга метаданных представляет собой объединение (или набор) модулей доступа метаданных.

При передаче услуги метаданных в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, каждой такой услуге присваивается уникальный идентификатор услуги метаданных. Идентификатор услуги метаданных уникальным образом идентифицирует услугу метаданных среди всех услуг метаданных, доступных в одном Транспортном или Программном потоке, и он *не* уникален *только лишь* внутри потока метаданных. Идентификатор услуги метаданных используется для выделения услуги метаданных и всей информации, требуемой для ее декодирования.

Для декодирования метаданных может потребоваться наличие данных о конфигурации декодера. Если для декодирования услуги метаданных, передаваемой в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, требуются данные о конфигурации декодера, тогда данные о конфигурации декодера для этих метаданных должны передаваться в той же программы того же потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

В подразделе 2.12.2 рассматривается синхронизация метаданных, а в подразделе 2.12.3 приводится обзор инструментов, которые определены для транспортировки метаданных в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Использование доступных инструментов транспортировки определяется в § 2.12.4–§ 2.12.8, а в § 2.12.9 определяется сигнализация, связанная с метаданными. Наконец, модель декодера STD для декодирования метаданных определяется в § 2.12.10.

Поскольку может передаваться множество форм метаданных, очень важно сообщить как о точном формате и кодировании метаданных, так и семантическом содержании, которое передается в метаданных. Первое сообщается форматом метаданных, тогда как второе сообщается форматом приложения метаданных. Другими словами, формат метаданных содержит информацию о том, как должны быть декодированы эти метаданные, тогда как формат приложения метаданных содержит информацию о том, как использовать метаданные, в особенности, какое приложение использует эти метаданные. Такое деление важно, поскольку оно отделяет кодирование или воспроизведение метаданных от их значения, и, следовательно, дает возможность приложение быть не зависеть от средств передачи метаданных.

2.12.2 Модель оси времени метаданных

Метаданные могут обозначать коды времени, связанные с содержанием, например для обозначения начала сегмента содержания. Каждое указание времени, сделанное в метаданных, обозначает определенную ось времени метаданных содержания, определяемую реальным форматом метаданных и/или форматом приложения метаданных. Например, один формат (приложения) метаданных может использовать коды времени UTC, тогда как другой формат приложения метаданных может использовать коды времени SMPTE. Для обеспечения возможности передачи содержания в любое время в любой среде передачи, ожидается, что ось времени метаданных содержания будет определена, но не будет зависеть от транспортировки.

При передаче содержания и связанных с ним метаданных в потоках соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, должны поддерживаться ссылки на точное время из метаданных с содержанием. То же самое необходимо, если метаданные доставляются при помощи других средств. Для того чтобы этого достичь, в настоящей Спецификации предполагается модель оси времени, показанная на рисунке 2-7.

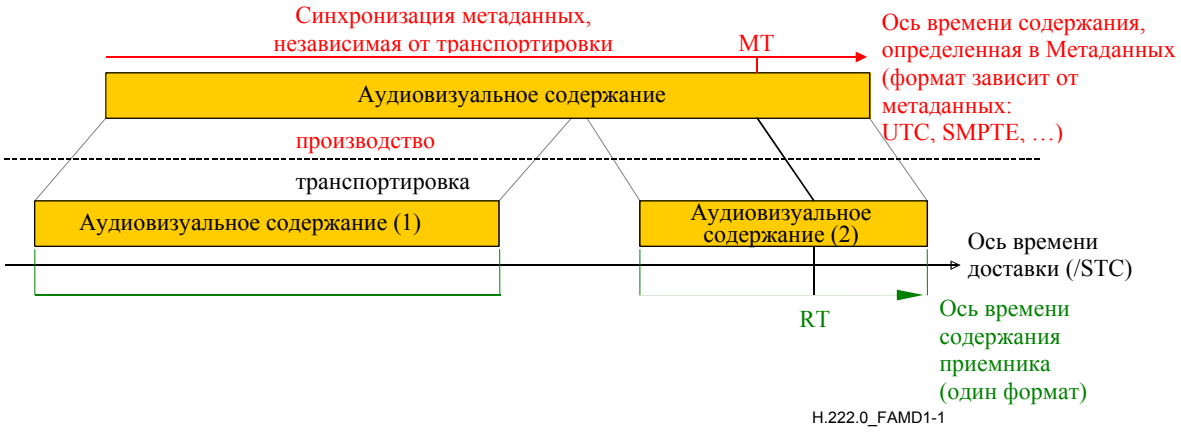


Рисунок 2-7 – Модель синхронизации для доставки контента и метаданных

Как правило, метаданные связываются с аудиовизуальным содержанием способом, который не зависит от транспортировки, на стадии производства или на любой другой стадии до транспортировки. При необходимости, для указания, например, определенных сегментов внутри содержания, в метаданные встраивается информация о времени, использующая ось времени метаданных содержания, которая применяется в метаданных. Например, могут использоваться коды времени UTC или SMPTE. Формат оси времени не зависит ни от одного кода времени, который может быть или не быть встроен в сам аудиовизуальный поток. Например, ось времени метаданных может использовать UTC, тогда как в видеопоток встроены коды времени SMPTE.

Для каждого потока метаданных должны выполняться следующие требования:

- не должна нарушаться непрерывность времени на оси времени метаданных содержания;
- ось времени метаданных содержания должна быть синхронизирована с частотой дискретизации содержания;
- каждая эталонная метка времени в потоке метаданных относится к одной и той же оси времени метаданных содержания.

При транспортировке, с содержанием связывается синхронизация, определяемая транспортировкой; она называется осью времени доставки. При транспортировке в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, ось времени доставки обеспечивается системными часами (STC). Содержание может быть доставлено в виде непрерывного блока информации, но также допускается прерывание доставки содержания, например в случае, когда программа прерывается экстренными сообщениями; в таком и некоторых иных случаях непрерывность времени может нарушаться.

Когда в метаданных используются метки времени, в декодере конечной системы (STD) эти метки времени должны быть однозначным образом связаны со значениями времени в принимаемом содержании. Для того чтобы этого достичь, требуется ось времени содержания приемника. В качестве оси времени содержания приемника могут использоваться системные часы (STC), но из-за нарушений непрерывности, которые могут происходить в STC, STC не обязательно обеспечит однозначную временную связь. Следовательно, в качестве оси времени содержания приемника может использоваться также концепция NPT (Нормальное время воспроизведения), описанная в ИСО/МЭК 13818-6 DSM-CC. В любом режиме воспроизведения – нормальном, обратном, замедленном, ускоренном, обратном ускоренном и неподвижных изображений, концепция NPT обеспечивает однозначную связь по времени, и она не зависит от вставок другого содержания. Отметим, что, когда STC возвращается к нулю, должен быть передан новый дескриптор NPT_reference_descriptor.

Для того чтобы поддерживать точные метки времени из метаданных с содержанием, требуется только информация о том, как преобразовать время метаданных (MT), определенное на оси времени метаданных содержания, в соответствующее время приемника (RT), на оси времени содержания приемника. Это выполняется за счет сдвига по времени (в единицах по 90 кГц) между осью времени метаданных содержания и осью времени содержания приемника. Этот сдвиг выполняется в дескрипторе разметки содержания. Этот сдвиг переносит в себе значение временной базы метаданных в момент времени, когда временная база содержания приемника достигает определенного значения. См. также рисунок 2-7.

Синхронизация в системах метаданных может ссылаться на определенный кадр изображения или звуковой кадр, например, используя коды времени SMPTE. Сдвиг по времени между осью времени метаданных содержания и осью времени содержания приемника выражается в единицах по 90 кГц, и, следовательно, в приемниках метка времени метаданных будет переведена в значение 90 кГц. Для того чтобы учесть неточности, приемник должен предположить, что когда ссылка сделана на кадр изображения или звуковой кадр, должны использоваться наиболее подходящие варианты. Например, переведенное значение метки времени метаданных в единицах по 90-кГц должно быть преобразовано в кадр изображения или звуковой кадр, значение PTS которого наиболее близко по значению с переведенным значением.

При использовании NPT, во время воспроизведения в любом режиме в любой момент времени сдвиг по времени между временной базой метаданных и временной базой NPT остается постоянным. До тех пор пока не нарушается непрерывность STC и не встречаются вставки другого содержания, это утверждение остается в силе для сдвига по времени между временной базой метаданных и временной базой STC, но только в нормальном режиме воспроизведения. Для осей времени, определенных в конфиденциальном порядке, сдвиг также должен оставаться постоянным, но, возможно, в рамках ограничений, которые не определены в настоящей Спецификации.

Когда применяется синхронная транспортировка метаданных в пакетах PES или за счет применения синхронного протокола загрузки DSM-CC, метаданным назначаются метки времени программы (PTS). Такие метки PTS могут, например, указывать момент времени, с которого метаданные становятся действующими. Это предполагает знание *априори* о том, как связать метаданные с синхронизацией доставки. Однако синхронно транспортируемые метаданные также могут содержать метки времени, которые должны быть преобразованы из оси времени метаданных содержания в ось времени содержания приемника с использованием определенного сдвига между обеими осями времени. См. также рисунок 2-8.

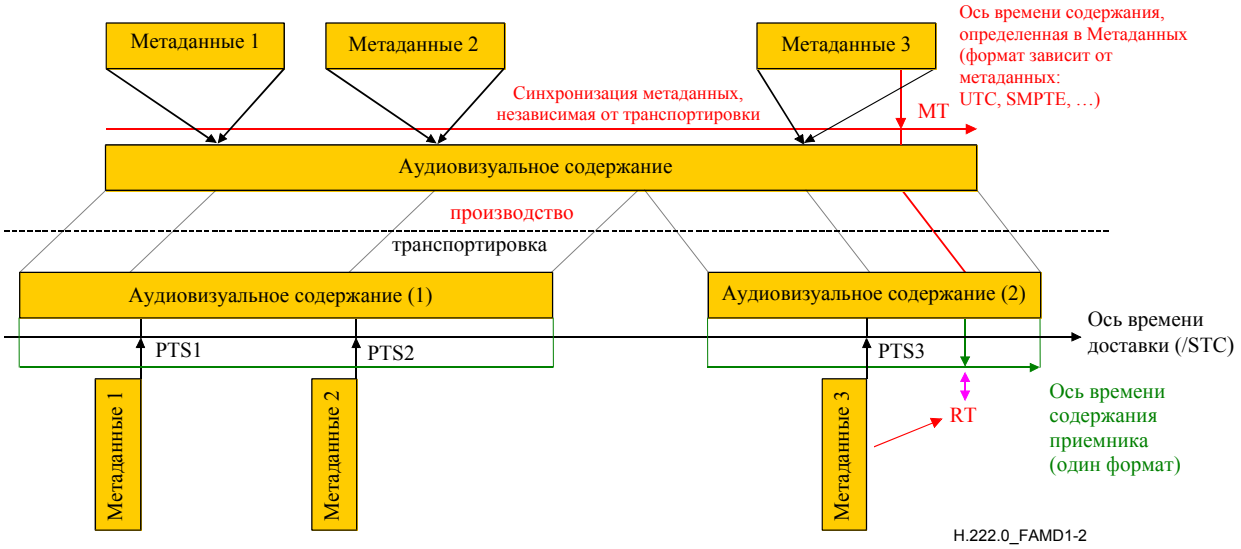


Рисунок 2-8 – Доставка метаданных в пакетах PES

2.12.3 Варианты транспортировки метаданных

Для того чтобы учесть разнообразные характеристики метаданных, определены различные методы транспортировки метаданных в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

В настоящей Спецификации определяется два метода синхронной доставки метаданных:

- передача в пакетах PES;
- использование протокола синхронизированной загрузки DSM-CC.

Кроме того, в настоящей Спецификации определяется три метода асинхронной доставки метаданных:

- передача в секциях метаданных;
- использование данных карусельного типа DSM-CC;
- использование объектов карусельного типа DSM-CC.

Отметим, что некоторые из вариантов асинхронной транспортировки поддерживают структуры карусельного типа и файловые структуры. Выбор метода транспортировки зависит от требований, которые предъявляются к доставке метаданных, и требований методов, которые описаны в последующих подразделах.

Метаданные могут также передаваться конфиденциальными средствами, такими как пакеты PES, имеющие значение идентификатора потока 0xBD или 0xBF (private_stream_id_1 или private_stream_id_2) или конфиденциальные секции. В настоящей Спецификации не определяется, каким образом для передачи метаданных используются конфиденциальные средства, но она позволяет сообщать о наличии таких метаданных, используя дескрипторы, определенные § 2.6.56–§ 2.6.63.

Базовое указание услуг метаданных одинаково для всех методов, использующих идентификаторы услуги метаданных. Однако существуют различия между методами. Когда используются пакеты PES, секции метаданных или синхронизированные секции загрузки DSM-CC, о данных каждой услуги метаданных явным образом сообщается в потоке метаданных, используя поле metadata_service_id. Однако, когда используются карусельные конструкции DSM-CC, такая сигнализация остается на усмотрение приложений метаданных. Отметим, что настоящая Спецификация допускает передачу услуги метаданных в карусельной конструкции DSM-CC, но не ограничивает количество услуг метаданных, которые могут передаваться в одной карусельной конструкции DSM-CC.

Данные метаданных о конфигурации декодера передаются в явном виде, когда они передаются в дескрипторе метаданных, в пакетах PES с полем stream_type = 0x15 и полем stream_id = 0xFC, в секциях метаданных или в синхронизированных секциях загрузки DSM-CC. Когда данные метаданных о конфигурации декодера передаются в карусельной конструкции DSM-CC, о таких данных требуется сообщать, но это не определяется настоящей Спецификацией; вместо этого, такая сигнализация остается на усмотрение приложений.

2.12.4 Использование пакетов PES для транспортировки метаданных

Пакеты PES предоставляют механизм для синхронной транспортировки метаданных. При помощи меток PTS в заголовке пакета PES модули доступа метаданных связываются с определенным моментом STC без необходимости меток времени в метаданных. Это предполагает знание *априори* о том, как связать метаданные с синхронизацией доставки. Сигнальным пакетам PES, в которых передаются метаданные, присваиваются определенные значения stream_id и stream_type; см. § 2.12.9.

Когда для транспортировки метаданных используются пакеты PES, имеющие stream_type = 0x15 и stream_id = 0xFC, в качестве метода выравнивания пакетов PES и модулей доступа метаданных, использующих ячейки metadata_AU_cell, должна использоваться Оболочка модуля доступа метаданных. Это даст возможность использовать индикацию случайного доступа, значение которого зависит от формата метаданных, и счетчик последовательности ячеек для идентификации потери ячеек metadata_AU_cell. Каждый модуль доступа метаданных передается и, при необходимости, разделяется на одну или несколько ячеек metadata_AU_cell. В каждом пакете PES, в котором передаются метаданные, первый байт PES_packet_data_byte должен быть первым байтом ячейки Metadata_AU_cell. Для каждого модуля доступа метаданных, содержащихся в одном пакете PES, применяется метка PTS в заголовке PES. Метка PTS сообщает время, в которое модули доступа метаданных мгновенно декодируются и удаляются из буфера Bn в декодере STD. Отметим, что в настоящей Спецификации не рассматривается соотношение между декодированным модулем доступа метаданных и аудиовизуальным содержанием.

Пакет PES может содержать одну ячейку metadata_AU_cell. Это удобно, если модуль доступа метаданных не может поместиться в одном пакете PES, в этом случае выполняется фрагментация метаданных модуля доступа метаданных Модуль при помощи ячейки metadata_AU_cell.

Когда метаданные передаются в Программном потоке при помощи пакетов PES, и, если в этом Программном потоке используется карта Программного потока, то карта Программного потока должна определить, какие пакеты PES содержат соответствующие метаданные.

2.12.4.1 Оболочка модуля доступа метаданных

Оболочка модуля доступа метаданных должна использоваться, когда модули доступа метаданных передаются в пакетах PES, имеющих значения stream_type = 0x15 и stream_id = 0xFC или в синхронизированных секциях загрузки DSM-CC, имеющих stream_type = 0x19. Оболочка определяет структуру, состоящую из множества объединенных ячеек metadata_AU_cell. Путем кодирования размера метаданных, содержащихся в каждой ячейке metadata_AU_cell, в приемниках может быть организован независимый синтаксический анализ метаданных. Анализатор может выделять метаданные и передавать их на декодер метаданных, не зная ничего об этих метаданных *априори*. Ячейка metadata_AU_cell должна быть синхронизирована с транспортировкой; то есть, первый байт полезной нагрузки пакета PES или синхронизированной секции загрузки DSM-CC должен быть первым байтом ячейки metadata_AU_cell.

Если модуль доступа метаданных в ячейку metadata_AU_cell полностью не помещается, то модуль доступа метаданных должен быть разделен на несколько ячеек metadata_AU_cell, в которых поле fragmentation_indication в каждой такой ячейке metadata_AU_cell сообщает, что данная ячейка metadata_AU_cell содержит фрагмент.

К каждой ячейки metadata_AU_cell, которая находится в том же пакете PES или в синхронизированной секции загрузки DSM-CC, применяется метка PTS, закодированная в заголовке пакета PES или синхронизированной секции загрузки DSM-CC, соответственно.

Таблица 2-96 – Оболочка модуля доступа метаданных

Синтаксис	Количество битов	Мнемоника
Metadata_AU_wrapper () {		
for (i = 0; i < N; i++){		
Metadata_AU_cell ()		
}		
}		

Таблица 2-97 – Ячейка модуля доступа метаданных

Синтаксис	Количество битов	Мнемоника
Metadata_AU_cell () {		
metadata_service_id	**8**	**uimsbf**
sequence_number	**8**	**uimsbf**
cell_fragment_indication	**2**	**bslbf**
decoder_config_flag	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
зарезервировано	**4**	**bslbf**
AU_cell_data_length	**16**	**uimsbf**
for (i = 0; I < AU_cell_data_length; i++){		
AU_cell_data_byte	**8**	**bslbf**
}		
}		

metadata_service_id: Это 8-битовое поле идентифицирует услугу метаданных, связанную с модулем доступа метаданных, передаваемым в этой ячейке модуля доступа метаданных.

sequence_number: Это 8-битовое поле определяет порядковый номер ячейки metadata_AU_cell. Этот номер увеличивается на единицу для каждой следующей ячейки metadata_AU_cell, образующей оболочку metadata_AU_wrapper, независимо от кодированного значения metadata_service_id.

cell_fragment_indication: Это 2-битовое поле передает информацию о модуле доступа метаданных, передаваемом в этой ячейке metadata_AU_cell, в соответствии с таблицей 2-98.

Таблица 2-98 – Указание фрагмента ячейки

Значение	Описание
11	Отдельная ячейка, в которой передается полный модуль доступа метаданных.
10	Первая ячейка в серии ячеек с данными из одного модуля доступа метаданных.
01	Последняя ячейка в серии ячеек с данными из одного модуля доступа метаданных.
00	Ячейка из серии ячеек с данными из одного модуля доступа метаданных, но не первая и не последняя ячейка серии

random_access_indicator: Это 1-битовое поле, которое, когда имеет значение '1', указывает, что метаданные, передаваемые в этой ячейке metadata_AU_cell, представляют собой точку входа для услуги метаданных, в которой возможно осуществление декодирования без информации из предыдущих ячеек metadata_AU_cell. Значение точки случайного доступа определяется форматом метаданных.

decoder_config_flag: Это 1-битовое поле сообщает о наличии в передаваемом модуле доступа метаданных информации о конфигурации декодера. Отметим, что это не препятствует возможности наличия метаданных в модуле доступа, соседнем с данными о конфигурации декодера.

AU_cell_data_length: Это 16-битовое поле определяет число байтов данных AU_cell_data_byte, которые следуют сразу после него.

AU_cell_data_byte: Это 8-битовое поле содержит непрерывную последовательность смежных байтов из модуля доступа метаданных.

2.12.5 Использование протокола синхронизированной загрузки DSM-CC для транспортировки метаданных

Для синхронизированной транспортировки, в дополнение к пакетам PES, может использоваться протокол синхронизированной загрузки DSM-CC. Когда для транспортировки метаданных используются синхронизированные секции загрузки DSM-CC, оболочка модуля доступа метаданных, определенная в § 2.12.4.1. должна использоваться как метод инкапсуляции модулей доступа метаданных. Это позволит использовать указание случайного доступа, значение которого зависит от формата метаданных, и счетчик последовательности ячеек для определения потери ячеек metadata_AU_cell. В каждой секции синхронизированной загрузки DSM-CC, в которой передаются метаданные, первый байт полезной нагрузки должен быть первым байтом ячейки metadata_AU_cell. Для каждого модуля доступа метаданных, содержащегося в одной секции синхронизированной загрузки DSM-CC, используется метка PTS в заголовке секции. Эта метка PTS сообщает время, в которое модули доступа метаданных мгновенно декодируются и удаляются из буфера B_n в декодере STD. Отметим, что в настоящей Спецификации не рассматривается соотношение между декодированным модулем доступа метаданных и аудиовизуальным содержанием. Для сообщения о передаче метаданных в синхронизированных секциях загрузки DSM-CC назначается специальное значение stream_type (подробно показанное в таблице 2-34).

2.12.6 Использование секций метаданных для транспортировки метаданных

Если требуется асинхронная транспортировка модулей доступа метаданных без карусельного механизма доставки, могут использоваться секции метаданных. Синтаксис и семантика секций метаданных определяются в настоящем подразделе. В каждой секции метаданных должен передаваться либо один полный модуль доступа метаданных, либо одна часть одного модуля доступа метаданных, это сообщается в поле section_fragment_indication.

Для транспортировки в секциях метаданных, модули доступа метаданных структурируются в виде одной или нескольких таблиц метаданных. Каждая таблица метаданных содержит один или несколько полных модулей доступа метаданных из одной или нескольких услуг метаданных. Концептуально, механизм транспортировки таблиц метаданных сравним с механизмом транспортировки таблиц карт программы или таблиц программных связей. Каждая таблица метаданных может быть создана из множества секций метаданных. Каждая таблица метаданных может содержать метаданные из нескольких услуг метаданных.

Секциям метаданных присваиваются специальные значения stream_type и table_id. В секциях также могут передаваться данные метаданных о конфигурации декодера, о чем сообщается значением Описание метаданных, присвоенным при помощи дескриптора конфигурации декодера метаданных.

Таблица 2-99 – Синтаксис секции для транспортировки метаданных

Синтаксис	Количество битов	Мнемоника
Metadata_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
random_access_indicator	**1**	**bslbf**
decoder_config_flag	**1**	**bslbf**
metadata_section_length	**12**	**uimsbf**
metadata_service_id	**8**	**uimsbf**
зарезервировано	**8**	**bslbf**
section_fragment_indication	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
for (i = 1; i < N; i++){		
metadata_byte	**8**	**bslbf**
}		
CRC_32	**32**	**rpchof**
}		

table_id: Идентификатор table_id – это 8-битовое поле, которое для каждой секции метаданных должно быть установлено равным '0x06'.

section_syntax_indicator: Это 1-битовое поле должно быть установлено равным '1'.

private_indicator: Это 1-битовое поле в настоящей Спецификации не определяется.

random_access_indicator: Это 1-битовое поле, которое, когда закодировано со значением '1', указывает, что метаданные, передаваемые в этой секции метаданных, представляют собой точку доступа к услуге метаданных, в которой возможно осуществление декодирования без информации из предыдущих секций метаданных. Значение точки случайного доступа определяется форматом метаданных.

decoder_config_flag: Это 1-битовое поле, которое, когда закодировано со значением '1', указывает, что в модуле доступа метаданных, передаваемом в этой секции метаданных, присутствует информация о конфигурации декодера.

metadata_section_length: Это 12-битовое поле должно определять число оставшихся байтов в секции, которые следуют сразу после поля metadata_section_length, включая CRC. Значение этого поля не должно превышать 4093 (0xFFD).

metadata_service_id: Это 8-битовое поле идентифицирует услугу метаданных, связанную с модулем доступа метаданных, передаваемым в этой секции метаданных. Каждая таблица метаданных может содержать метаданные из нескольких услуг метаданных.

section_fragment_indication: Это 2-битовое поле содержит информацию о фрагментации модуля доступа метаданных, передаваемого в этой секции метаданных, в соответствии с таблицей 2-100.

Таблица 2-100 – Указание фрагмента секции

Значение	Описание
11	Отдельная секция метаданных, в которой передается полный модуль доступа метаданных.
10	Первая секция метаданных в серии секций метаданных с данными из одного модуля доступа метаданных.
01	Последняя секция метаданных в серии секций метаданных с данными из одного модуля доступа метаданных.
00	Секция метаданных из серии секций метаданных с данными из одного модуля доступа метаданных, но не первая и не последняя секция серии.

version_number: Это 5-битовое поле представляет собой номер версии полной таблицы метаданных. Номер версии должен увеличиваться на 1 (по модулю 32) всегда, когда изменяется информация, содержащаяся в таблице метаданных. Когда индикатор current_next_indicator установлен в '1', значение version_number должно соответствовать текущей применяемой таблице метаданных. Когда индикатор current_next_indicator установлен в '0', значение version_number должно соответствовать следующей применяемой таблице метаданных.

current_next_indicator: 1-битовое поле, которое, когда установлено в '1', указывает, что в настоящее время применяется переданная таблица метаданных. Когда этот бит установлен в ‘0’, он указывает, что переданная таблица метаданных более не применима и должна вступить в действие следующая таблица метаданных.

section_number: Это 8-битовое поле содержит номер секции метаданных. Поле section_number первой секции в таблице метаданных должно иметь значение 0x00. Поле section_number должно увеличиваться на 1 с появлением в этой таблице метаданных каждой дополнительной секции.

last_section_number: Это 8-битовое поле определяет номер последней секции (то есть, секции с наибольшим значением поля section_number) в полной таблице метаданных, к которой принадлежит эта секция.

metadata_byte: Это 8-битовое поле указывает число непрерывных байтов из модуля доступа метаданных.

CRC_32: Это 32-битовое поле должно содержать значение CRC, которое дает ноль на выходе регистров декодера, определенного в Приложении A, после обработки полной секции metadata_section.

2.12.7 Использование данных карусельного типа DSM-CC для транспортировки метаданных

Могут использоваться методы DSM-CC, определенные для данных карусельного типа в ИСО/МЭК 13818-6, если требуется механизм доставки карусельного типа, без необходимости выражать в механизме транспортировки иерархическую структуру организации метаданных. Информация о карусельной структуре, в которой содержатся метаданные, вводится в дескриптор метаданных, определенный в § 2.6.60 и § 2.6.62. Для сообщения о передаче метаданных в данных карусельного типа DSM-CC назначается специальное значение stream_type. Отметим, что в данных карусельного типа DSM-CC требуется сигнализация об услугах метаданных, но она не определяется в настоящей Спецификации.

2.12.8 Использование объекта карусельного типа DSM-CC для транспортировки метаданных

Если требуется механизм доставки карусельного типа с возможностью выражения иерархической организации структуры метаданных при транспортировке, то могут использоваться методы DSM-CC и файловые структуры, которые определены в ИСО/МЭК 13818-6 для объектов карусельного типа "пользователь – пользователь". Эти файловые структуры предоставляют методы для структурирования метаданных в соответствии с тем, как представляется приемлемым для эффективного синтаксического анализа метаданных и для выражения иерархической организации метаданных. Информация, необходимая для идентификации карусели, в которой содержатся метаданные, включается в дескриптор метаданных, определенный в § 2.6.60 и § 2.6.61. Она может представлять собой IOP:IOR(), как определено в § 11.3.1 и § 5.7.2.3 стандарта ИСО/МЭК 13818-6 DSM-CC. Для сигнализации передаче метаданных в объекте карусельного типа DSM-CC назначается особое значение поля stream_type. Отметим, что услуга сигнализация о передаче метаданных внутри объекта карусельного типа DSM-CC требуется, но в настоящей Спецификации она не определяется.

2.12.9 Сигнализация, связанная с метаданными

Сигнализация, связанная с метаданными охватывает четыре четко различимых области:

- услуги и потоки сигнализации о метаданных;
- сигнальная передача содержания, предназначенного для использования системой метаданных;
- связь метаданных с содержанием; и
- сигнальная передача данных о конфигурации декодера.

2.12.9.1 Услуги и потоки сигнализации о метаданных

О передаче метаданных сообщается при помощи значения stream_type, находящегося в диапазоне между 0x15 и 0x19, которое определяет, какой из пяти методов, описанных в § 2.12.4–§ 2.12.8, используется для транспортировки метаданнных.

Для того чтобы уникальным образом идентифицировать услугу метаданных, каждой такой услуге при транспортировке присваивается значение идентификатора metadata_service_id; это назначенное значение должно быть уникальным в пределах Транспортного или Программного потока, в котором передается эта услуга метаданных. Если метаданные передаются в пакетах PES, имеющих stream_id = 0xFC, или в секциях метаданных, или в синхронизированных секциях загрузки, соответствующих ИСО/МЭК 13818-6, то назначенное значение metadata_service_id передается явным образом в заголовке ячейки metadata_AU_cell или секции метаданных. Если для передачи метаданных используется карусель, соответствующая ИСО/МЭК 13818-6, то сигнализация об услугах метаданных выполняется приложением. Дескриптор метаданных определяет формат метаданных и предоставляет информацию о конфигурации декодера и связан с услугой метаданных путем передачи информации об услуге метаданных, с которой он связан.

2.12.9.2 Сигнальная передача содержания, предназначенного для использования системой метаданных

В § 2.6.56 и § 2.6.57 определен дескриптор маркировки содержания, который может использоваться для назначения аудиовизуальному или любому другому содержанию, передаваемому в Транспортном или Программном потоке MPEG-2, специальной записи формата приложения метаданных content_reference_id_record. Запись content_reference_id_record может использоваться системой метаданных в качестве метки для указания на это содержания. Содержание может быть представлено, например, программой или потоком, или их сегментами. Кроме того, дескриптор маркировки содержания содержит информацию о временной базе содержания, используемой для определения эталонных моментов времени при помощи метаданных, включая постоянный сдвиг во времени между временной базой метаданных и временной базой применяемого содержания. Этот дескриптор позволяет передавать конфиденциальную информацию. Формат приложения метаданных metadata_application_format может определять ограничения для записи content_reference_record, например, ограничения периода времени, в течение которого эта запись остается действительной.

2.12.9.3 Связь метаданных с содержанием

В § 2.6.58 и § 2.6.59 определен дескриптор указателя метаданных, предназначенный для связи отдельной услуги метаданных с аудиовизуальным или иным содержанием в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Метаданные связываются с содержанием в рамках контекста, как определено местоположением дескриптора. В Транспортном потоке дескриптор может располагаться в цепочке дескриптора PMT либо для программы, либо для элементарного потока, но может также располагаться и в таблицах, которые не определены в настоящей Спецификации, например, в таблицах, описывающих услуги групповой или радиовещательной передачи.

Дескриптор указателя метаданных указывает из контекста содержания на услугу метаданных, связанную с этим содержанием. Дескриптор содержит значение идентификатора metadata_service_id, который назначен связанной с ним услуге метаданных, а также данные об одном или нескольких местоположениях соответствующих метаданных. Местоположение может находиться, например, внутри того же Транспортного потока, где расположено содержание, либо в другом Транспортном потоке, но также и представлять собой местоположение в потоке, не соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, таком как интернет.

2.12.9.4 Сигнальная передача данных о конфигурации декодера

Для декодирования метаданных может потребоваться доступность метаданных о конфигурации декодера. Если они требуются, то данные о конфигурации декодера должны содержаться в одной из услуг метаданных в той же программе в том же потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, в котором находится эта услуга метаданных. Если данные о конфигурации декодера требуются для декодирования услуги метаданных, то дескриптор метаданных либо содержит эти данные, либо содержит информацию о том, как получить данные о конфигурации декодера из той же или иной услуги метаданных. В Транспортном потоке такую другую услугу можно найти в результате поиска в таблице PMT дескриптора metadata_descriptor, имеющего идентификатор metadata_service_id, который определен в поле decoder_config_metadata_service_id (и который имеет такой же формат метаданных metadata_format и такой же формат приложения метаданных metadata_application).

2.12.9.5 Обзор сигнальной передачи метаданных

На рисунке 2-9 показан пример сигнальной передачи метаданных, в котором отдельная программа ("программа содержания") передает содержание (или смысловую часть), а метаданные передаются в отдельной программе, ("программе метаданных"). В этом примере программа метаданных и программа содержания существуют в одном Транспортном потоке.

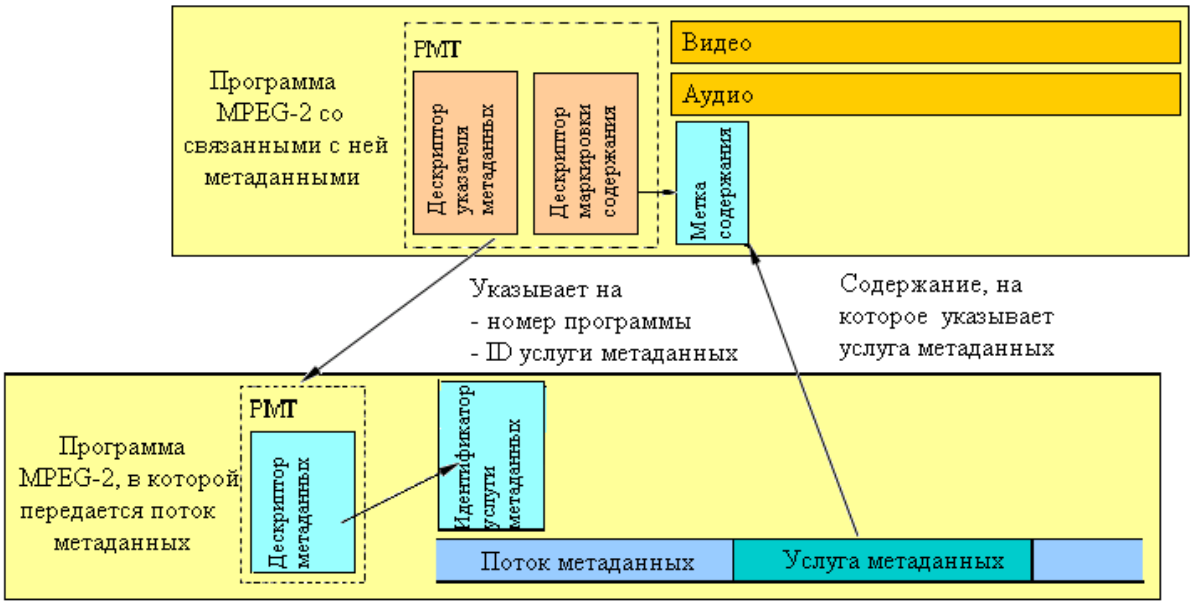


Рисунок 2-9 – Сигнальная передача и указание метаданных

В программе содержания находится два дескриптора, связанных с метаданными – дескриптор маркировки содержания content_labeling descriptor и дескриптор указателя метаданных metadata_pointer descriptor. Дескриптор content_labeling descriptor связывает с содержанием метку, показанную на рисунке как "метка содержания" и закодированную в полях content_reference_id дескриптора. Эта метка может использоваться услугой метаданных для обозначения содержания, либо в целом, либо частично, либо при помощи сегмента, который описан указаниями времени. Например, дескриптор content_labeling может содержать метку "News of 1/1/02", и тогда метаданные могут указывать на конкретный блок сюжета внутри "News of 1/1/02", например, путем указания конкретного момента времени этого блока сюжета.

Дескриптор указателя метаданных содержит информацию о том, где может располагаться услуга метаданных для данного содержания. В этом примере, метаданные передаются в отдельной программе, но равнозначно действительным может быть наличие метаданных, передаваемых как в той же программе, что и содержание, так и предоставляемых другими средствами, выходящими за рамки настоящей Спецификации, например, полученными их URL. Этот дескриптор, также, содержит значение идентификатора услуги метаданных, который назначен услуге метаданных. Это требуется, поскольку поток метаданных может содержать несколько услуг метаданных для множества различных программ, и каждая программа должна иметь возможность уникальным образом идентифицировать свою собственную услугу метаданных.

В программе метаданных дескриптор метаданных сообщает, к какой услуге метаданных внутри данного потока метаданных он относится. Дескриптор метаданных, если он используется, содержит подробную информацию о том, где находится информация о конфигурации декодера.

После идентификации дескриптора указателя метаданных в PMT при помощи декодирования программы содержания в приемнике, приемник выделяет из программы метаданных дескриптор метаданных. При необходимости, сначала выделяются данные о конфигурации декодера, затем соответствующим образом конфигурируется декодер, после чего может быть начато декодирование услуги метаданных.

2.12.10 Модель декодера STD для метаданных

Модель декодера STD определяет нормативные ограничения для потоков, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, в которых передаются метаданные. Для декодирования метаданных в декодере STD применяются обычные модели декодеров T-STD и P-STD со скоростью ввода метаданных в буфер $B_n = Rx_n$ и скоростью вывода метаданных из буфера $B_n = R_{metadata}$, которая также равна скорости ввода метаданных в декодер $D_{metadata}$. См. рисунок 2-10.

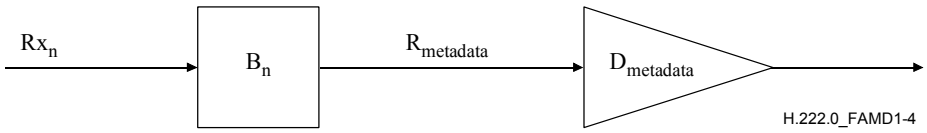


Рисунок 2-10 – Декодирование метаданных в декодере STD

Метаданные поступают в буфер B_n со скоростью Rx_n. В декодере P-STD скорость Rx_n равна скорости Программного потока. В декодере T-STD скорость Rx_n равна скорости выхода из TB_n и равна скорости, определенной в поле metadata_input_leak_rate дескриптора метаданных декодера STD. Размер BS_n буфера B_n равен размеру, определенному в поле metadata_buffer_size дескриптора метаданных декодера STD. В случае синхронной доставки, декодирование метаданных выполняется синхронно и управляется метками времени PTS. В момент времени декодирования, то есть, когда STC = PTS, соответствующие метаданные мгновенно удаляются из B_n. В случае асинхронной доставки, метаданные удаляется из B_n со скоростью $R_{metadata}$, равной скорости, определенной в поле metadata_output_leak_rate в дескрипторе декодера метаданных STD. Буфер B_n не должен переполняться.

Отметим, что модель декодера STD определяет ограничения на доставку метаданных, без указания каких-либо ограничений на синхронизацию, используемую в метаданных.

2.13 Передача данных, соответствующих ИСО 15938

2.13.1 Введение

Передача метаданных в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, определенная в § 2.12, позволяет передавать данные, соответствующие с ИСО 15938, при помощи соответствующего кодирования поля metadata_format. В настоящем подразделе, для целей транспортировки данных, соответствующих ИСО 15938, определяется особый элемент. Передача данных, соответствующих ИСО 15938, должна удовлетворять требованиям, определенным в § 2.12, но, кроме того, требования, определенные в настоящем подразделе, должны применяться для транспортировки данных, соответствующих ИСО 15938.

2.13.2 Данные о конфигурации декодера ИСО 15938

Для декодирования данных, соответствующих ИСО 15938, требуется доступность данных о конфигурации декодера. Следовательно, когда данные ИСО 15938 передаются в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, дескриптор метаданных должен сообщать о передаче соответствующих данных о конфигурации декодера в том же потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, за счет кодирования значения флага decoder_config_flag либо '001', либо '010', либо '011', либо '100'.

2.14 Передача видеопотока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10

2.14.1 Введение

В настоящей спецификации определяется передача элементарного потока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, в системах, описываемых Рекомендацией МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, как для Программных, так и для Транспортных потоков. Как правило, поток, соответствующий Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, является элементом программы ИСО/МЭК 13818-1, как определено таблицей PMT в Транспортном потоке и картой PSM в Программном потоке. Передача и управление буферами видеопотоков AVC определяется с использованием существующих параметров из данной Рекомендации | Международного стандарта, таких как PTS и DTS, а также информации, содержащейся в видеопотоке AVC.

Передача видеопотоков AVC в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, определяет обеспечить точное соответствие между параметрами STD и параметрами HRD, которые могут присутствовать в видеопотоке AVC. Требования определены для случая, когда в видеопотоке AVC присутствуют параметры HRD, что гарантирует возможность проверить, соблюдаются ли все требования STD для каждого видеопотока AVC, передаваемого в составе Транспортного или Программного потока.

ПРИМЕЧАНИЕ 1. – Хотя в информации о синхронизации, содержащейся с видеопотоке AVC, таймер с частотой 90 кГц может не использоваться, метки времени PTS и DTS должны выражаться в единицах по 90 кГц.

Если поток, соответствующий Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, передается в составе потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, то данные, закодированные в соответствии с Рекомендацией МСЭ-Т H.264 | ИСО/МЭК 14496-10, должны содержаться в пакетах PES. В данных, кодированных в соответствии с Рекомендацией МСЭ-Т H.264 | ИСО/МЭК 14496-10, должен соблюдаться формат потока байтов, определенный в Приложении B к Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, со следующими ограничениями:

- Каждый модуль доступа AVC содержит разделитель модуля доступа NAL;

  ПРИМЕЧАНИЕ 2. – Рекомендация МСЭ-Т H.264 | ИСО/МЭК 14496-10 требует, чтобы разделитель модуля доступа блока NAL, если он имеется, был бы первым блоком NAL в модуле доступа AVC. Разделители модулей доступа в блоках NAL упрощают определение границы между изображениями. Они позволяют избежать необходимости обрабатывать содержимое заголовков секций, и они особенно полезны для базового и расширенного профилей, в которых порядок следования секций может быть произвольным.

- Каждый блок NAL в потоке байтов, содержащий разделитель модулей доступа, должен иметь только один синтаксический элемент zero_byte.

  ПРИМЕЧАНИЕ 3. – Синтаксис и семантика блоков NAL в потоке байтов определены в Приложении B Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10.

- В видеопотоке должны присутствовать все наборы параметров последовательностей и изображений (SPS и PPS), необходимые для декодирования видеопотока AVC.

  ПРИМЕЧАНИЕ 4. – Рекомендация МСЭ-Т H.264 | ИСО/МЭК 14496-10 также позволяет передавать SPS и PPS при помощи внешних средств. Данная спецификация не обеспечивает поддержку такой передачи и поэтому требует, чтобы SPS и PPS передавались в составе видеопотока AVC.

- Каждая видеопоследовательность AVC, содержащая набор параметров hrd_parameters() с флагом low_delay_hrd_flag, установленным в '1', должна содержать параметры VUI, в которых флаг timing_info_present_flag установлен в '1'.

  ПРИМЕЧАНИЕ 5. – Если флаг low_delay_hrd_flag установлен в '1', то неполное заполнение буфера в модели декодера STD является допустимым; см. § 2.14.3 и § 2.14.4. Установка флага timing_info_present_flag в '1' гарантирует, что видеопоток AVC содержит достаточно информации для определения времени выхода из буфера декодированных изображений (DPB) и времени удаления из буфера кодированных изображений (CPB) в модуле доступа AVC, в том числе и в случае исчерпания буфера.

Для того чтобы обеспечить отображение конкретной информации, такой как, например, aspect_ratio (соотношение сторон кадра), настоятельно рекомендуется, чтобы в каждом видеопотоке AVC передавались параметры использования визуальной информации (VUI) с данными, достаточными для того, чтобы гарантировать, что декодированный видеопоток AVC может быть корректно воспроизведен на приемниках.

2.14.2 Передача в составе пакетов PES

Видеоданные, соответствующие Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, передаются в составе пакетов PES как данные полей PES_packet_data_bytes с использованием одного из 16 значений идентификатора stream_id, присвоенного видеоинформации, при этом о передаче видеопотока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, сигнализируется посредством назначенного значения поля stream_type в таблице PMT или карте PSM (см. таблицу 2-29). Наивысший уровень, который может встретиться в видеопотоке AVC, а также профиль, которому соответствует поток в целом, должны указываться с использованием видеодескриптора AVC. Если дескриптор AVC связан с видеопотоком AVC, то он должен быть включен в цепочку дескрипторов для соответствующей записи элементарного потока в таблице карты программы (в случае Транспортного потока) или в карте программного потока, если присутствует PSM (в случае Программного потока). Данная Рекомендация | Международный стандарт не определяют воспроизведение потоков, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, в контексте программы.

При разбивке на пакеты PES никаких специальных ограничений для выравнивания данных не применяется. Для синхронизации и управления декодером STD, метки PTS и, если допустимо, DTS, кодируются в заголовке пакета PES, в котором передаются данные элементарного видеопотока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Ограничения и семантика, применяемые при кодировании меток PTS и DTS, определены в § 2.4.3.7 и § 2.7.

2.14.3 Расширения декодера STD

2.14.3.1 Расширения декодера T-STD

Модель T-STD включает в себя транспортный буфер TB_n и буфер мультиплексирования MB_n,, предшествующий буферу EB_n для декодирования каждого элементарного видеопотока "n", соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. См. рисунок 2-11.

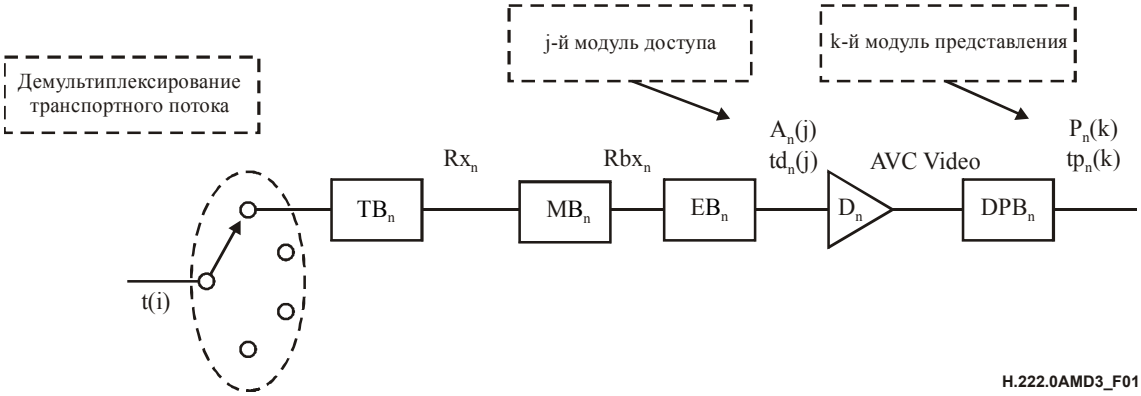


Рисунок 2-11 – Расширения модели декодера T-STD для видеоданных, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10

Управление буфером DPB_n

Передача видеопотока AVC в потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, не влияет на размер буфера DPB_n. Для декодирования видеопотока AVC в декодере STD размер буфера DPB_n соответствует размеру, определенному в Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10. Управление буфером DPB осуществляется, как описано в Приложении C к Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 (§ C.2 и § C.4). Декодированный модуль доступа AVC поступает в буфер DPB_n сразу же после завершения декодирования, т. е. в момент удаления модуля доступа AVC из буфера кодированных изображений (CPB). Декодированный модуль доступа AVC появляется в момент времени выхода из буфера декодированных изображений (DPB). Если видеопоток AVC не содержит достаточной информации для определения времени удаления модулей доступа AVC из буфера CPB и времени выхода модулей доступа AVC из буфера DPB, то эти моменты времени должны быть определены в модели декодера STD на основе меток времени PTS и DTS следующим образом:

1) Время удаления модуля доступа AVC "n" из буфера CPB представляет собой момент времени, определяемый меткой DTS("n"), где DTS("n") – значение метки DTS модуля доступа AVC "n".

2) Время выхода модуля доступа AVC "n" из буфера DPB представляет собой момент времени, определяемый меткой PTS("n"), где PTS("n") – значение метки PTS модуля доступа AVC "n".

ПРИМЕЧАНИЕ 1. – Видеопоследовательности AVC, в которых флаг low_delay_hrd_flag в наборе параметров hrd parameters() установлен в ‘1’, содержат достаточную информацию, для того чтобы определить для каждого модуля доступа AVC время выхода из DPB и время удаления из буфера CPB. Следовательно, в модулях доступа AVC, для которых может произойти недозаполнение декодера STD, время удаления из буфера CPB и время выхода из буфера DPB определяются параметрами HRD, а не метками времени DTS и PTS.

Управление буферами TB_n, MB_n и EB_n

Входной сигнал буфера TB_n и его размер TBS_n описаны в § 2.4.2.3. Для буферов MB_n и EB_n, и для скорости Rx_n передачи потока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, между буферами TB_n и MB_n и скорости Rbx_n передачи этого потока данных между буферами MB_n и EB_n соблюдаются следующие ограничения:

Размер EBS_n буфера EB_n:

EBS_n = cpb_size,

где cpb_size – размер буфера CPB (CpbSize[cpb_cnt_minus1]) для формата потока байтов, определяемого в параметрах NAL hrd_parameters(), которые передаются в составе параметров VUI в видеопотоке AVC. Если в видеопотоке AVC параметры NAL hrd_parameters() отсутствуют, то cpb_size является величиной, определенной как 1200 × MaxCPB в Приложении A Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 для уровня видеопотока AVC.

Размер MBS_n буфера MB_n:

$$MBS_n = BS_{mux} + BS_{oh} + 1200 \times MaxCPB[level] - cpb_size$$

где BS_{oh}, – величина буфера для заголовка пакета, определяется как:

BS_{oh} = (1/750) секунд × max{1200 × MaxBR[level], 2 000 000 бит/секунду},

а дополнительный буфер мультиплексирования BS_{mux} определяется как:

BS_{mux} = 0,004 секунд × max{1200 × MaxBR[level], 2 000 000 бит/секунду },

где MaxCPB[level] и MaxBR[level] определены для формата потока байтов в таблице A.1 (Ограничения уровня) в Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 для уровня видеопотока AVC, и

где cpb_size – размер буфера CPB (CpbSize[cpb_cnt_minus1]) для формата потока байтов, определяемого в параметрах NAL hrd_parameters(), которые передаются в составе параметров VUI в видеопотоке AVC. Если в видеопотоке AVC отсутствуют параметры NAL hrd_parameters(), то cpb_size является величиной, определенной как 1200 × MaxCPB в Приложении A Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 для уровня видеопотока AVC.

Скорость Rx_n:

Если в TB_n, нет данных, то Rx_n равна нулю.

В ином случае: Rx_n = bit_rate,

где bit_rate – скорость потока битов данных BitRate[cpb_cnt_minus1] в CPB для формата потока байтов, определяемого в параметрах NAL hrd_parameters(), которые передаются в составе параметров VUI в видеопотоке AVC. Если параметры NAL hrd_parameters() отсутствуют в видеопотоке AVC, то bit_rate является скоростью потока битов, определенной в Приложении A Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10 для уровня видеопотока AVC как 1200 × MaxBR[level].

Передача между MB_n и EB_n

Если и дескриптор синхронизации AVC, и HRD присутствуют, и флаг hrd_management_valid_flag установлен в '1', то передача данных из буфера MB_n в буфер EB_n должна соответствовать определенной схеме приема данных в буфер CPB декодера HRD, как определено в Приложении C Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10.

В ином случае, для передачи данных между буферами MB_n и EB_n используется метод утечки следующим образом:

Скорость Rbx_n:

$$Rbx_n = 1200 \times MaxBR[level]$$

где MaxBR[level] определяется для формата потока байтов на каждом уровне в таблице A.1 (ограничения уровня) Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10.

Если в буфере MB_n имеются данные полезной нагрузки пакета PES, и буфер EB_n не заполнен, то полезная нагрузка пакета PES передается из буфера MB_n в буфер EB_n со скоростью, равной Rbx_n. Если буфер EB_n полон, то данные из буфера MB_n не удаляются. Если байт данных передан из буфера MB_n в буфер EB_n, то все байты заголовка пакета PES, находящиеся в MB_n и предшествующие переданному байту, немедленно удаляются и сбрасываются. Если в MB_n нет данных полезной нагрузки пакета PES, то никакие данные из буфера MB_n не удаляются. Все данные, поступающие в буфер MB_n, покидают его. Все байты данных полезной нагрузки пакета PES сразу же после выхода из буфера MB_n поступают в буфер EB_n.

Удаление модулей доступа AVC из EB_n

Каждый модуль доступа AVC $A_n(j)$, присутствующий в буфере EB_n, удаляется немедленно в момент $td_n(j)$. Время декодирования $td_n(j)$ определяется меткой времени DTS или на основе времени удаления из буфера CPB, получаемого из данных видеопотока AVC.

Задержка в декодере STD

Общая задержка в буферах оконечных системных декодеров TB_n, MB_n и EB_n любых данных, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, отличных от неподвижного изображения AVC, ограничивается величиной $td_n(j) - t(i) \leq 10$ секунд для всех значений j и для всех i-тых байтов в модуле доступа AVC $A_n(j)$.

Задержка данных любых неподвижных изображений AVC в буферах TB_n, MB_n и EB_n оконечных системных декодеров ограничивается величиной $td_n(j) - t(i) \leq 60$ секунд для всех значений j и для всех i-тых байтов в модуле доступа AVC $A_n(j)$.

Условия управления буферами

Транспортные потоки должны быть составлены таким образом, чтобы выполнялись следующие условия управления буфером:

- буфер TB_n не должен переполняться и должен опустошаться хотя бы один раз в течение каждой секунды.
- буферы MB_n, EB_n и DPB_n не должны переполняться.

- буфер EB_n не должен быть недозаполненным, кроме случая, когда в видеопоследовательности AVC, у которой флаг low_delay_hrd_flag установлен в '1', присутствуют параметры VUI. Недозаполнение буфера EB_n имеет место для модуля доступа AVC $A_n(j)$, когда во время декодирования $td_n(j)$ в EB_n не присутствуют один или несколько байтов $A_n(j)$.

ПРИМЕЧАНИЕ 2. – Видеопоток AVC может содержать информацию для определения соответствия параметров потока HRD, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, как определено в Приложении C. О присутствии этой информации в транспортном потоке говорит использование дескриптора синхронизации AVC и HRD с флагом hrd_management_valid_flag, установленным в '1'. Независимо от наличия этой информации соответствие видеопотока AVC параметрам декодера T-STD гарантирует, что выполняются требования управления буфером HRD для буфера CPB_n, если каждый байт в видеопотоке AVC доставляется в буфер CPB_n и удаляется из него в составе HRD именно в тот самый момент времени, когда этот байт доставляется или удаляется из буфера EB_n в декодере T-STD.

2.14.3.2 Расширения декодера P-STD

Модель P-STD для декодирования элементарного потока, соответствующего Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, включает в себя буфер мультиплексирования B_n и декодер D_n, за которым следует буфер DPB_n (см. рисунок AMD3-2). Для каждого видеопотока AVC "n" размер BS_n буфера B_n в P-STD определяется значением поля P-STD_buffer_size в заголовке пакета PES.

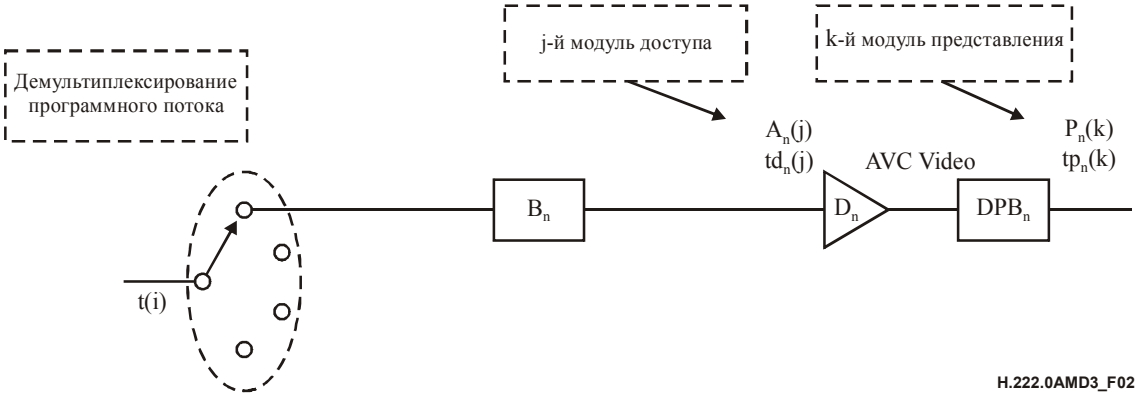


Рисунок 2-12 – Расширения модели декодера P-STD для видеоданных, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10

Управление буфером DPB_n

Управление буфером DPB_n осуществляется так же, как и в случае декодера T-STD; см. п. 2.14.3.1.

Управление буфером B_n

Данные модуля доступа AVC поступают в буфер B_n, как описано в § 2.5.2.2. В момент времени $td_n(j)$ модуль доступа AVC $A_n(j)$ декодируется и немедленно удаляется из буфера B_n. Время декодирования $td_n(j)$ задается меткой времени DTS или моментом времени удаления из буфера CPB, данные о котором извлекаются из информации видеопотока AVC. После декодирования модуль доступа AVC немедленно поступает в буфер DPB_n или поступает на выход, минуя DPB_n, в соответствии с правилами, определенными в Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10.

Задержка декодера STD

Общая задержка в буфере конечного системного декодера B_n данных, соответствующих Рекомендации МСЭ-Т H.264 | ИСО/МЭК 14496-10, отличных от неподвижных изображений AVC, ограничивается величиной $td_n(j) - t(i) \leq 10$ секунд для всех значений j и для всех i-тых байтов в модуле доступа AVC $A_n(j)$.

Задержка любых неподвижных изображений AVC в буфере конечного декодера системны B_n ограничивается величиной $td_n(j) - t(i) \leq 60$ секунд для всех j и всех байтов модуля доступа AVC $A_n(j)$.

Условия управления буфером

Программные потоки должны быть составлены таким образом, чтобы выполнялись следующие условия управления буфером:

- буфер B_n не должен переполняться.
- буфер B_n не должен быть недозаполненным, кроме случая, когда в видеопоследовательности AVC, у которой флаг low_delay_hrd_flag установлен в '1', присутствуют параметры VUI, или когда состояние параметра trick_mode = "true". Неполное заполнение буфера B_n имеет место для модуля доступа AVC $A_n(j)$, когда во время декодирования $td_n(j)$ в EB_n отсутствует один или несколько байтов $A_n(j)$.

Приложение A

Модель декодера CRC

(Настоящее Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

A.0 Модель декодера CRC

32-битовая модель декодера CRC определена на рисунке A.1.

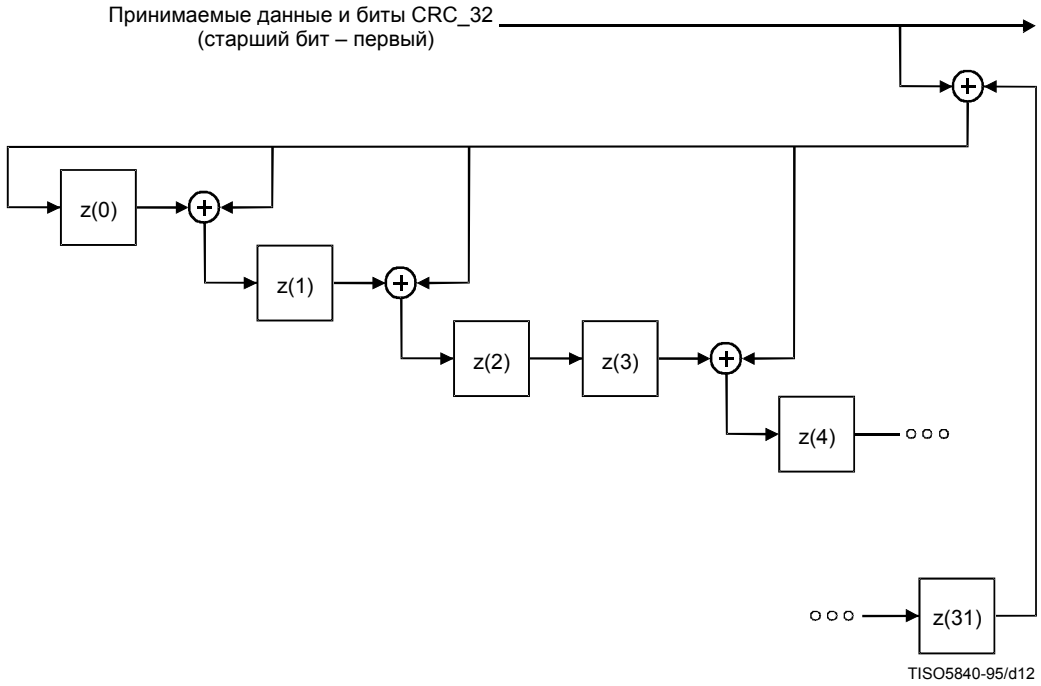


Рисунок A.1 – 32-битовая модель декодера CRC

32-битовый декодер CRC действует на уровне битов и состоит из 14 сумматоров '+' и 32 элементов задержки z(i). Входной сигнал декодера CRC добавляется на выход z(31), и полученный результат подается на вход z(0) и на один из входов каждого оставшегося сумматора. Другой вход каждого оставшегося сумматора является входом z(i), тогда как выход каждого оставшегося соединяется с входом z(i + 1), где i = 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22 и 25. См. рисунок A.1, выше.

Это значение CRC, рассчитанное с помощью полинома:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1 \qquad \text{(A-1)}$$

Байты принимаются на входе декодера CRC. Каждый байт побитно сдвигается внутрь декодера CRC, при этом первым является самый левый бит (старший). Например, если входным сигналом является байт 0x01, то сначала в декодер CRC поступает семь '0', и после них одна '1'. Прежде, чем CRC начнет обрабатывать данные секции, выходной сигнал каждого элемента задержки z(i) устанавливается в свое исходное значение '1'. После такого инициирования каждый байт секции поступает на вход декодера CRC, включая четыре байта CRC_32. После сдвига последнего бита последнего байта CRC_32 внутрь декодера, т. е. внутрь z(0) после сложения с выходным сигналом z(31), считывается выходной сигнал со всех элементов задержки. В том случае, когда ошибок нет, каждый из выходных сигналов z(i) должен быть равен нулю. В кодере CRC поле CRC_32 кодируется с таким значением, чтобы гарантировалось выполнение этого требования.

Приложение B

Управление и контроль для цифровой среды хранения данных (DSM-CC)

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

B.0 Введение

Протокол DSM-CC представляет собой специальный прикладной протокол, предназначенный для выполнения базовых функций и операций управления, характерных для управления двоичным потоком, соответствующим Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, на цифровом запоминающем устройстве. Настоящий протокол DSM-CC является протоколом низкого уровня, расположенного выше уровней сети/OS и ниже прикладных уровней.

DSM-CC должен быть прозрачным в следующем смысле:

- Он не зависит от используемой цифровой среды хранения данных (DSM);
- Он не зависит от того, расположена ли DSM на местном или удаленном сайте;
- Он не зависит от сетевого протокола, с которым стыкуется DSM-CC;
- Он не зависит от различных операционных систем, на которых действует DSM.

B.0.1 Цель

В приложениях, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 DSM, для многих команд управления требуется доступ к двоичному потоку, соответствующему Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, сохраненному в различных цифровых запоминающих устройствах на местном или удаленном сайте. В различных DSM имеются свои собственные команды управления, и, таким образом, для того чтобы получить доступ к двоичному потоку, соответствующему Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, пользователь будет вынужден знать различные наборы специальных команд управления DSM. Это приводит к возникновению множества сложностей при разработке интерфейса для системы приложений в соответствии с Рекомендацией МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 или ИСО/МЭК 11172-1. Для преодоления этой сложности, в настоящем Приложении предлагается использовать набор общих команд управления DSM, который не зависят от конкретной применяемой системы DSM. Настоящее Приложение является только информативным. В стандарте ИСО/МЭК 13818-6 определяется расширение DSM-CC с более широкой сферой применения.

B.0.2 Будущие приложения

Кроме приложений, поддерживаемых существующими командами управления DSM, будущие приложения, построенные на основе расширений команд управления DSM, могут включать в себя следующее:

Видео по запросу

Видеопрограммы по запросу пользователя предоставляются по различным каналам связи. Пользователь может выбрать видеопрограмму из перечня программ, доступных на видеосервере. Такие приложения могут применяться в гостиницах, сетях кабельного ТВ, учебных заведениях, больницах и т. п.

Интерактивные видеослужбы

В таких приложениях пользователь часто выполняет действия, управляя сохраненными видео- и аудио-материалами. Эти услуги могут включать в себя видеоигры, видеопутешествия, управляемые пользователем, электронный шоппинг и т. п.

Видеосети

В различных приложениях может потребоваться обеспечить передачу сохраненной видео- и аудио-информацией по компьютерной сети какого-либо типа. Пользователи могут направлять аудио и видео информацию по видеосети на свои терминалы. Примерами этого типа приложений являются электронные издательские системы и мультимедийные приложения.

B.0.3 Преимущества

Формируя спецификации команд управления DSM, не зависящие от цифровой среды передачи (DSM), конечные пользователи могут выполнять декодирование в соответствии с Рекомендацией МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 без необходимости полного понимания подробностей работы конкретной используемой DSM.

Команды управления DSM представляют собой коды, которые дают пользователю гарантию того, что Двоичные потоки, соответствующие Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, могут быть воспроизведены и сохранены с неизменной семантикой, вне зависимости от DSM и интерфейса пользователя. Это базовые команды управления работой среды хранения данных (DSM).

B.0.4 Базовые функции

B.0.4.1 Выбор потока

Команды DSM-CC предоставляют возможность выбрать двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, для которого должны выполняться последующие операции. Эти операции включают в себя создание нового двоичного потока. Параметры этой функции включают в себя:

- индекс двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, (преобразование этого индекса в название, имеющее смысл для приложения, в существующей DSM не рассматривается);
- режим (извлечение/сохранение).

B.0.4.2 Извлечение

Команды DSM-CC предоставляют возможность:

- воспроизведения указанного двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1;
- воспроизведения, начиная с указанного времени воспроизведения;
- установить скорость воспроизведения (нормальное или ускоренное);
- установить продолжительность воспроизведения (до определенного времени воспроизведения, окончание двоичного потока при воспроизведении вперед или начало потока при обратном воспроизведении или появление команды "стоп");
- установить направление (прямое или обратное);
- пауза;
- возобновление;
- изменение точки доступа в двоичном потоке;
- стоп.

B.0.4.3 Хранение

Команды DSM-CC предоставляют возможность:

- вызова операции сохранения действующего двоичного потока на протяжении указанного времени;
- вызова операции сохранения до команды "стоп".

Команды DSM-CC предоставляют чрезвычайно полезный, но ограниченный набор функций, которые могут требоваться в приложениях DSM, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Вполне можно ожидать, что посредством дополнительных расширений будут добавлены существенные дополнительные возможности.

B.1 Общие элементы

B.1.1 Сфера применения

Сфера интересов настоящей заключается в разработке Рекомендации | Международного стандарта для определения полезного набора команд для управления цифровым запоминающим устройством, в котором записывается двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Эти команды могут осуществлять дистанционное управление цифровым запоминающим устройством, в общем случае, независимо от конкретной DSM и применительно к любому сохраненному в DSM двоичному потоку, соответствующему Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

B.1.2 Обзор приложения DSM-CC

Существующие синтаксис и семантика DSM-CC относятся к приложению DSM для отдельного пользователя. Система пользователя способна извлечь двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, и, также (дополнительно) способна сформировать двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Канал управления, по которому передаются команды и подтверждения DSM, показан на рисунке B.1 в виде внеполосного канала. Если внеполосный канал недоступен, то эти операции могут выполняться также путем введения команд и подтверждений DSM-CC в двоичные потоки, соответствующие Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

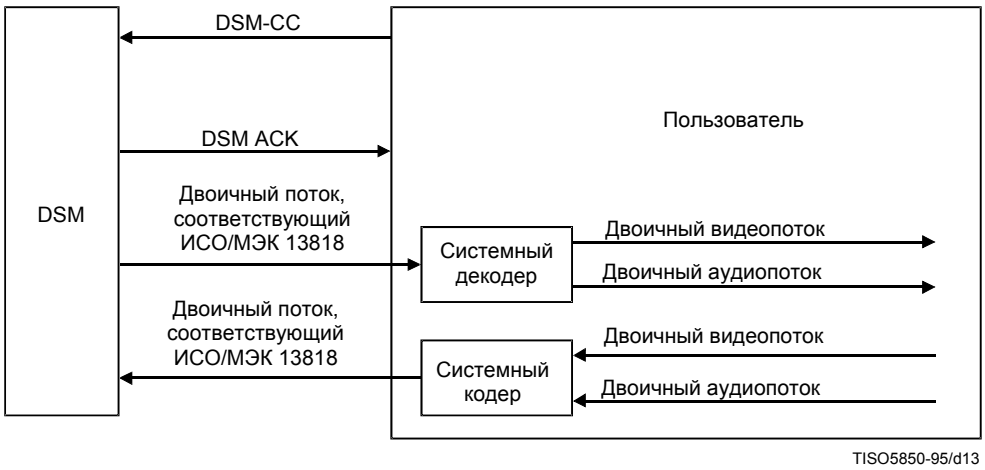


Рисунок B.1 – Конфигурация приложения DSM-CC

B.1.3 Передача команд и подтверждений DSM-CC

Команда управления DSM-CC кодируется в виде двоичного потока DSM-CC в соответствии с синтаксисом и семантикой, определенными в § B.2.2–§ B.2.9. Двоичный поток DSM-CC может быть передан как в виде отдельного двоичного потока, так и в виде системного двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

Когда двоичный поток DSM-CC передается в режиме отдельного двоичного потока, его взаимосвязь с системным двоичным потоком и процессом декодирования показан на рисунке B.2. В таком случае, двоичный поток DSM-CC не вводится в системный двоичный поток. Этот режим передачи может использоваться в тех приложениях, где DSM непосредственно связана с декодером, соответствующим Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Он может использоваться также в тех приложениях, где двоичный поток DSM-CC может управляться и передаваться через другие типы сетевых мультиплексоров.

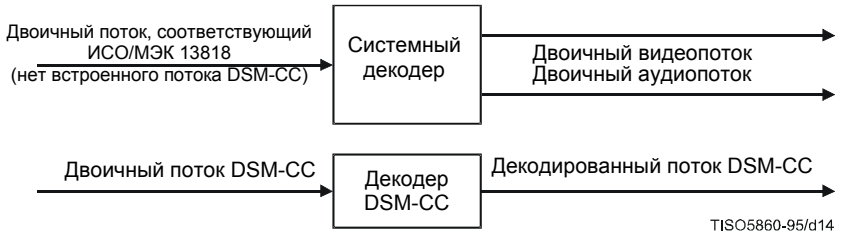


Рисунок B.2 – Двоичный поток DSM-CC, декодированный как отдельный двоичный поток

Для некоторых приложений желательно передавать сигналы DSM-CC в системном двоичном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, так что некоторые возможности системного двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, могут также применяться и к двоичному потоку DSM-CC. В таком случае, двоичный поток DSM-CC вводится в системный двоичный поток при помощи системного мультиплексора.

DSM-CC кодируется системным кодером следующим образом. Сначала, двоичный поток DSM-CC разбивается на пакеты и формируется элементарный поток пакетов (PES) в соответствии с синтаксисом, описанным в § 2.4.3.6. Затем поток PES мультиплексируется, формируя либо Программный поток (PS), либо Транспортный поток (TS) в соответствии с требованиями среды передачи. Процедуры декодирования обратны процедурам кодирования и показаны на блок-диаграмме системного декодера, изображенной на рисунке B.3.

На рисунке B.3 выходной сигнал системного декодера представляет собой двоичный видеопоток, двоичный аудиопоток и/или двоичный поток DSM-CC. Двоичный поток DSM-CC идентифицируется значением stream_id = '1111 0010'. Как определено в таблице идентификаторов потоков (stream_id) – таблице 2-22. После того, как двоичный поток DSM-CC идентифицирован, он следует правилам, установленным декодером T-STD или P-STD.

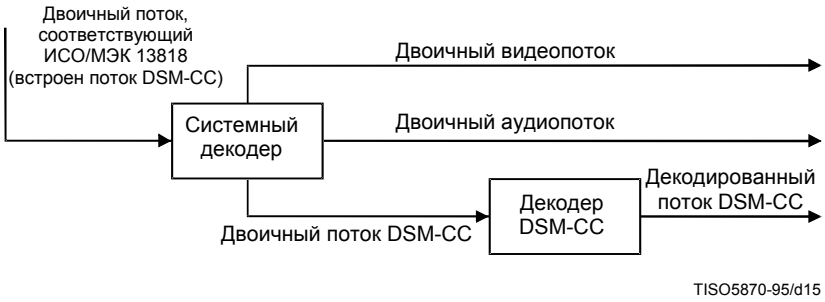


Рисунок B.3 – Двоичный поток DSM-CC, декодированный как часть системного потока

B.2 Технические элементы

B.2.1 Определения

В тексте настоящей Рекомендации | Международного стандарта, применяются следующие определения:

B.2.1.1 DSM-CC: Команды контроля и управления цифрового запоминающего устройства, которые определены Рекомендацией МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 для управления цифровым запоминающим устройством на местном или удаленном сайте, содержащие двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

B.2.1.2 DSM ACK: Подтверждение, передаваемое от приемника команды DSM-CC инициатору команды.

B.2.1.3 Двоичный поток MPEG: Системный поток, соответствующий ИСО/МЭК 11172-1, Программный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, или Транспортный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

B.2.1.4 Сервер DSM-CC: Система, местная или удаленная, используемая для хранения и/или извлечения двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

B.2.1.5 Точка случайного доступа: точка в двоичном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, с такими свойствами, что в данном двоичном потоке, как минимум, для одного элементарного потока, следующий модуль доступа ('N'), полностью содержащийся в данном двоичном потоке, может быть декодирован без ссылки на предыдущие модули доступа, и для каждого элементарного потока в данном двоичном потоке все модули доступа с тем же или более поздними значениями времени воспроизведения полностью содержатся в данном двоичном потоке, и могут быть полностью декодированы декодером конечной системы без доступа к информации, находящейся до точки случайного доступа. Двоичный поток, сохраненный в DSM, может иметь определенные точки случайного доступа; выходной сигнал DSM может содержать дополнительные точки случайного доступа, созданные собственными действиями системы DSM над сохраненным материалом (например, услуги сохранения квантования, так что заголовок последовательности может быть сформирован в любой время, когда это необходимо). Точка случайного доступа имеет связанную с ней метку PTS, а именно, реальную или подразумеваемую метку PTS модуля доступа 'N'.

B.2.1.6 текущее рабочее значение метки PTS: Реальная или подразумеваемая метка PTS, связанная с последней точкой случайного доступа, предшествующей последнему модулю доступа, полученному от DSM из выбранного в настоящее время двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Если из этого двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, не получено ни одного модуля доступа, то DSM не способна предоставить случайный доступ в текущий двоичный поток, тогда текущим рабочим значением метки PTS является первая точка случайного доступа в двоичном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

B.2.1.7 Двоичный поток DSM-CC: Последовательность битов, удовлетворяющая синтаксису § B.2.2.

B.2.2 Спецификация синтаксиса DSM-CC

- Каждая команда управления DSM должна начинаться со стартового кода start_code, определенного в таблице B.1.
- Каждая команда управления DSM должна содержать поле packet_length, определяющее число битов в пакете DSM-CC.
- Когда двоичный поток DSM-CC передается в виде пакетов PES, как определено в § 2.4.3.6, все поля до поля packet_length идентичны тем полям, которые определены в § 2.4.3.6. Другими словами, если пакет DSM-CC инкапсулирован в пакет PES, то стартовым кодом пакета PES является единственный стартовый код в начале пакета.

- Действующая команда управления или подтверждение должна следовать после последнего байта поля packet_length.
- Поток подтверждений должен формироваться приемников двоичного потока команд управления DSM после того, как запрошенное действие начато или завершено, в зависимости от принятой команды.
- На всех этапах DSM несет ответственность за формирование нормативного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Эти действия могут предусматривать работу с битами "хитрого" режима, определенными в § 2.4.3.6.

Таблица B.1 – Синтаксис DSM-CC

Синтаксис	Количество битов	Мнемоника
DSM_CC() {		
packet_start_code_prefix	**24**	**bslbf**
stream_id	**8**	**uimsbf**
packet_length	**16**	**uimsbf**
command_id	**8**	**uimsbf**
If (command_id = = '01') {		
control()		
} else if (command_id = = '02') {		
ack()		
}		
}		

B.2.3 Семантика полей в спецификации синтаксиса DSM-CC

packet_start_code_prefix – Это 24-битовый код. Вместе с идентификатором потока stream_id, который следует после него, он образует стартовый код пакета DSM-CC, который обозначает начало двоичного потока пакетов DSM-CC. Префикс packet_start_code_prefix – это строка битов '0000 0000 0000 0000 0000 0001' (0x000001).

stream_id – Это 8-битовое поле определяет тип двоичного потока и для двоичного потока DSM-CC должно иметь значение '1111 0010'. См. таблицу 2-23.

packet_length – Это 16-битовое поле определяет количество байтов в пакете DSM-CC, который следует сразу после последнего байта этого поля.

command_id – Это 8-битовое целое число без знака, идентифицирует двоичный поток, который является потоком команд управления или подтверждений. Его значения определяются в таблице B.2.

Таблица B.2 – Назначенные значения идентификатора command_id

Значение	Command_id
0x00	Запрещенное значение
0x01	Управление
0x02	Подтверждение
0x03-0xFF	Зарезервированное значение

B.2.4 Уровень управления

Ограничения на установку флагов в управлении DSM-CC

- В каждой команде управления DSM, значение '1' может быть установлено не более чем для одного флага "выбор", "воспроизведение" и "хранение". Если ни один из этих битов не установлен, то эта команда должна игнорироваться.
- В каждой команде извлечения потока значение '1' может быть установлено не более чем для одного флага pause_mode, resume_mode, stop_mode, play_flag и jump_flag. Если ни один из этих битов не установлен, то эта команда должна игнорироваться.
- Для каждой команды сохранения может быть выбрано не более одного флага record_flag и stop_mode. Если ни один из этих битов не установлен, то эта команда должна игнорироваться.

См. таблицу B.3.

Таблица B.3 – Управление DSM-CC

Синтаксис	Количество битов	Мнемоника
control() {		
select_flag	**1**	**bslbf**
retrieval_flag	**1**	**bslbf**
storage_flag	**1**	**bslbf**
зарезервировано	**12**	**bslbf**
marker_bit	**1**	**bslbf**
If (select_flag = = '1') {		
bitstream_id [31..17]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
bitstream_id [16..2]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
bitstream_id [1..0]	**2**	**bslbf**
select_mode	**5**	**bslbf**
marker_bit	**1**	**bslbf**
}		
if (retrieve_flag = = '1') {		
jump_flag	**1**	**bslbf**
play_flag	**1**	**bslbf**
pause_mode	**1**	**bslbf**
resume_mode	**1**	**bslbf**
stop_mode	**1**	**bslbf**
зарезервировано	**10**	**bslbf**
marker_bit	**1**	**bslbf**
if (jump_flag = ='1') {		
зарезервировано	**7**	**bslbf**
direction_indicator	**1**	**bslbf**
time_code()		
}		
if (play_flag = ='1'){		
speed_mode	**1**	**bslbf**
direction_indicator	**1**	**bslbf**
зарезервировано	**6**	**bslbf**
time_code()		
}		
}		
if (storage_flag = ='1') {		
зарезервировано	**6**	**bslbf**
record_flag	**1**	**bslbf**
stop_mode	**1**	**bslbf**
if (record_flag = ='1') {		
time_code()		
}		
}		
}		

B.2.5 Семантика полей на уровне управления

marker_bit – Это 1-битовый маркер, который всегда установлен в '1' во избежание эмуляции стартового кода.

reserved_bits – Это 12-битовое поле зарезервировано для будущего использования в настоящей Рекомендации | Международном стандарте для команд управления DSM . До тех пор, пока в документах МСЭ-Т | ИСО/МЭК не будет определено иное, оно должно иметь значение '0000 0000 0000'.

select_flag – Этот 1-битовый флаг, когда он установлен в '1', указывает операцию выбора потока. Когда он установлен '0', операция выбора потока выполняться не должна.

retrieval_flag – Когда этот 1-битовый флаг установлен в '1', он определяет, что будет выполняться конкретное действие извлечения (воспроизведения). Это действие начинается с текущего рабочего значения метки времени PTS.

storage_flag – Когда этот 1-битовый флаг установлен в '1', он определяет, что будет выполняться операция сохранения.

bitstream_ID – Это 32-битовое поле кодируется в виде трех частей. Эти части объединяются, формируя целое число без знака, определяющее, какой двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, должен быть выбран. Сервер DSM выполняет преобразование названий сохраненных в его DSM двоичных потоков, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, в последовательность чисел, которая может быть представлена в виде идентификатора двоичного потока (bitstream_ID).

select_mode – Это 5-битовое целое число без знака определяет, какой режим работы двоичного потока запрашивается. Определенные спецификацией режимы показаны в таблице B.4.

Таблица B.4 – Назначенные значения для поля "выбор режима"

Код	Режим
0x00	Запрещенное значение
0x01	Хранение
0x02	Извлечение
0x03-0x1F	Зарезервированное значение

jump_flag – Когда этот 1-битовый флаг установлен в '1', он определяет переход в указателе воспроизведения на новый модуль доступа. Новая метка времени PTS определяется соответствующим кодом time_code по отношению к текущему рабочему значению PTS. Эта функция действительна только, когда текущий двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, находится в режиме "стоп".

play_flag – Когда этот 1-битовый флаг установлен в '1', он определяет воспроизведение двоичного потока в течение определенного периода времени. Скорость, направление и продолжительность воспроизведения являются дополнительными параметрами в двоичном потоке. Воспроизведение начинается с текущего рабочего значения PTS.

pause_mode – Это однобитовый код, определяющий паузу в процессе воспроизведения и устанавливающий указатель воспроизведения в текущее рабочее значение PTS.

resume_mode – Это однобитовый код, определяющий продолжение воспроизведения, начиная с текущего рабочего значения PTS. Возобновление воспроизведения имеет смысл только, если текущий двоичный поток находится в режиме "пауза", и двоичный поток будет установлен в режим прямого воспроизведения с нормальной скоростью.

stop_mode – Это однобитовый код, определяющий остановку передачи двоичного потока.

direction_indicator – Это однобитовый код, указывающий направление воспроизведения. Если этот бит установлен в ‘1', он означает прямое воспроизведение. В ином случае он означает воспроизведение в обратном направлении.

speed_mode – Это 1-битовый код, определяющий масштаб скорости. Если этот бит установлен в '1', он указывает нормальную скорость воспроизведения. Если этот бит установлен в '0', он указывает, что воспроизведение выполняется на высокой скорости (т. е. Ускоренное воспроизведение или Ускоренное обратное воспроизведение).

record_flag – Это однобитовый флаг, определяющий направленный на DSM запрос конечного пользователя записать двоичный поток в течение определенного времени или до получения команды "стоп", в зависимости от того, то случится раньше.

B.2.6 Уровень подтверждения

Ограничения на установку флагов в управлении DSM-CC

В каждом двоичном потоке подтверждений DSM только один из определенных ниже битов подтверждений может быть установлен в '1' (см. таблицу B.5).

Таблица B.5 –Подтверждение DSM-CC

Синтаксис	Количество битов	Мнемоника
ack() {		
select_ack	**1**	**bslbf**
retrieval_ack	**1**	**bslbf**
storage_ack	**1**	**bslbf**
error_ack	**1**	**bslbf**
зарезервировано	**10**	**bslbf**
marker_bit	**1**	**bslbf**
cmd_status	**1**	**bslbf**
If (cmd_status = = '1' && (retrieval_ack = ='1' || storage_ack = = '1')) { time_code() }		
}		

B.2.7 Семантика полей на уровне подтверждения

select_ack – Когда это 1-битовое поле установлено в '1', оно указывает, что команда ack() подтверждает получение команды "выбор".

retrieval_ack – Когда это 1-битовое поле установлено в '1', оно указывает, что команда ack() подтверждает получение команды "извлечение" .

storage_ack – Когда это 1-битовое поле установлено в '1', оно указывает, что команда ack() подтверждает получение команды "сохранение".

error_ack – Когда это 1-битовое поле установлено в '1', оно указывает ошибку DSM. Определенными ошибками являются EOF (окончание файла при прямом воспроизведении или начало файла при обратном воспроизведении) в извлекаемом потоке и Disk Full (диск заполнен) в сохраняемом потоке. Если этот бит установлен в '1', то состояние cmd_status не определено. Выбранным остается текущий двоичный поток.

cmd_status – Когда этот 1-битовый флаг установлен в '1', он указывает, что команда принята. Когда он установлен в '0', он указывает, что команда отброшена. Семантика меняется в соответствии с принятой командой следующим образом:

- Если установлено поле select_ack и поле cmd_status установлено в '1', это означает, что выбран двоичный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, и сервер готов предоставить выбранный режим работы. Текущее рабочее значение PTS устанавливается в первую точку случайного доступа нового выбранного двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Если поле cmd_status установлено в ‘0', это значит, что операция не выполнена и ни одного двоичного потока не выбрано.
- Если установлено поле retrieval_ack и поле cmd_status установлено в '1', это означает, что инициирована операция извлечения для всех команд извлечения. В следующем поле time_code сообщается положение текущего рабочего указателя метки PTS.
- Для команды play_flag, у которой флаг infinite_time_flag = '1', будет передано второе подтверждение. Оно подтвердит, что операция воспроизведения завершено по достижению продолжительности, определенной командой play_flag.
- Если в подтверждении извлечения поле cmd_status установлено в '0', это значит, что операция не выполнена. Возможными причинами неудачи могут быть: недействительный bitstream_ID, переход в точку, расположенную после окончания файла или неподдерживаемая функция, например, обратное воспроизведение со стандартной скоростью.

- Если установлено поле storage_ack, это означает, что операция сохранения для команды record_flag начинается или завершается по команде stop_mode. В следующем поле time_code сообщается метка времени PTS последнего полностью сохраненного модуля доступа.
- Если операция записи завершается по достижении и продолжительности, определенной командой storage_flag, то должно быть предано еще одно подтверждение, и должно быть сообщено текущее рабочее значение PTS после записи.
- Если в подтверждении сохранения поле cmd_status установлено в '0', это значит, что операция не выполнена. Возможными причинами неудачи могут быть: недействительный bitstream_ID или неспособность DSM сохранить данные.

B.2.8 Временной код

Ограничения временного кода

- Прямая операция определенной продолжительности, которая указана в поле time_code, завершается после того, как появляется реальная или предполагаемая метка времени PTS модуля доступа, так что результат "эта метка PTS минус текущее рабочее значение PTS в начале операции по модулю 2^{33}" превышает эту продолжительность.
- Обратная операция определенной продолжительности, которая указана в поле time_code, завершается после того, как появляется реальная или предполагаемая метка времени PTS модуля доступа, так что результат "текущее рабочее значение PTS в начале операции минус эта метка PTS по модулю 2^{33}" превышает эту продолжительность.
- Для всех команд на уровне control() поле time_code определяется как относительная продолжительность относительно текущего рабочего значения PTS.
- Для всех команд на уровне ack() поле time_code определяется текущим рабочим значением PTS.

См. таблицу B.6.

Таблица B.6 – Временной код

Синтаксис	Количество битов	Мнемоника
time_code() {		
зарезервировано	**7**	**bslbf**
infinite_time_flag	**1**	**bslbf**
if (infinite_time_flag = = '0') {		
зарезервировано	**4**	**bslbf**
PTS [32..30]	**3**	**bslbf**
marker	**1**	**bslbf**
PTS [29..15]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
PTS [14..0]	**15**	**bslbf**
marker_bit	**1**	**bslbf**
}		
}		

B.2.9 Семантика полей во временном коде

infinite_time_flag – Когда этот 1-битовый флаг установлен в '1', он указывает бесконечный период времени. Этот флаг установлен в '1' в тех приложениях, где период времени для определенной операции не может быть определен заранее.

PTS [32..0] – Метка воспроизведения модуля доступа двоичного потока. В зависимости от функции, она может иметь абсолютное или относительное значение задержки во времени, выраженное в циклах системных часов с частотой 90 кГц.

Приложение C

Специальная информация программы

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

C.0 Описание информации, относящейся к программе, в Транспортных потоках

В подразделе 2.4.4 содержатся нормативные синтаксис, семантика и текст, касающиеся специальной информации программы. Во всех случаях требуется соответствие ограничениям, указанным в п. 2.4.4. В настоящем Приложении содержится пояснительная информация о том, как использовать функции PSI, и рассматриваются примеры того, как она может использоваться на практике.

C.1 Введение

В настоящей Рекомендации | Международном стандарте содержится метод описания содержания пакетов Транспортного потока для целей демультиплексирования и воспроизведения программ. Спецификация кодирования включает в себя эту функцию с помощью Специальной информации программы (PSI). В настоящем Приложении рассматривается использование PSI.

Можно считать, что PSI находится в шести таблицах:

1) Таблице ассоциаций для программы (PAT);
2) Таблице преобразования программы TS (PMT);
3) Таблице сетевой информации (NIT);
4) Таблице условного доступа (CAT);
5) Таблице описания Транспортного потока; и
6) Таблице управляющей информации IPMP.

Содержание таблиц PAT, PMT, CAT и TSDT описано в настоящей Рекомендации | Международном стандарте. Дескриптор IPMP определен в стандарте ИСО/МЭК 13818-11 (MPEG-2 IPMP).

NIT – это конфиденциальная таблица, и значения идентификаторов (PID) пакетов Транспортного потока, в которых она передается, определены в PAT. Как таблица NIT, так и таблица ICIT должны соответствовать структуре разделов, определенной в настоящей Рекомендации | Международном стандарте.

C.2 Функциональный механизм

Все вышеперечисленные таблицы являются концептуальными, то есть в том плане, что они никогда не восстанавливаются в декодере в определенной спецификацией форме. Хотя эти структуры можно рассматривать как простые таблицы, до передачи их в пакетах Транспортного потока, они могут быть разделены на части. Синтаксис поддерживает эти действия, позволяя делить таблицы на секции, вводя в полезную нагрузку пакета Транспортного потока нормативный метод преобразования. Кроме того, предусматривается метод для передачи в аналогичном формате и конфиденциальной информации. Это имеет свои преимущества, поскольку одни и те же базовые операции обработки в декодере могут использоваться и для данных PS, и для конфиденциальной информации, что позволяет снизить стоимость. Рекомендации относительно оптимального места расположения PSI в Транспортном потоке приведены в Приложении D.

Каждая секция уникальным образом идентифицируется при помощи комбинации следующих элементов:

i) **table_id**

8-битовый идентификатор table_id показывает, к какой таблице относится раздел**.**

- Разделы с table_id = 0x00 принадлежат Таблице ассоциаций для программы.
- Разделы с table_id = 0x01 принадлежат Таблице условного доступа.
- Разделы с table_ id = 0x02 принадлежат Таблице преобразования программы TS.
- Разделы с table_id = 0x03 принадлежат разделу TS_ description_section.
- Разделы с table_id = 0x04 принадлежат разделу ISO_IEC_14496_scene_ description_section.
- Разделы с table_id = 0x05 принадлежат разделу ISO_IEC_14496_ object_descriptor_section.
- Разделы с table_id = 0x06 принадлежат разделу metadata_section.
- Разделы с table_id = 0x07 принадлежат разделу IPMP_Control_information_section.

Другие значения идентификатора table_id могут быть назначены пользователем для конфиденциальных целей.

Допускается установка фильтров, просматривающих поле table_id, с целью определения, принадлежит ли новая секция интересующей нас таблице или нет.

ii) **table_id_extension**

Это 16-битовое поле существует в длительной версии секции. В Таблице ассоциаций для программы оно используется для идентификации значения transport_stream_id потока. В действительности, это определяемая пользователем метка, которая позволяет отличить один Транспортный поток от другого внутри сети или среди различных сетей. В настоящее время это поле в Таблице условного доступа не имеет никакого значения и, следовательно, отмечено как "зарезервированное", это значит, что оно должно быть закодировано как 0xFFFF, но его значение может быть определено МСЭ-Т | ИСО/МЭК при последующем пересмотре настоящей Рекомендации | Международного стандарта. В секции преобразования программы TS это поле содержит номер программы program_number, и, следовательно, определяет программу, к которой относятся данные в этой секции. В некоторых случаях расширение table_id_extension может использоваться также как точка фильтрации.

iii) **section_number**

Поле section_number позволяет декодеру восстановить исходный порядок следования секций определенной таблицы. В настоящей Рекомендации | Международном стандарте не устанавливается никаких требований относительно того, что секции должны передаваться в последовательном порядке, но рекомендуется его придерживаться, если только нет необходимости предавать некоторые секции таблицы чаще остальных, например, из соображений случайного доступа.

iv) **version_number**

Когда меняются характеристики Транспортного потока, описанные в PSI, (например, добавляются дополнительные программы, иные композиции элементарных потоков для данной программы), то должны быть переданы новые данные PSI с обновленной информацией, так чтобы всегда была бы действительной та версия секций, отмеченных как "текущие", которая передана последней по времени. Декодеры должны иметь возможность определить, идентична ли секция, принятая последняя по времени, той секции, которая уже была обработана/сохранена (в этом случае секция должна быть отброшена), или это другая секции, и, следовательно, она может означать изменение конфигурации. Это обеспечивается за счет передачи секции с теми же значениями полей table_id, table_id_extension и section_number, что и в предыдущей секции, содержащими требуемые данные, но со следующим значением поля version_number.

v) **current_next_indicator**

Очень важно знать, в какой точке двоичного потока действительна данная PSI. Следовательно, каждая секция может быть обозначена как действительная "сейчас" (текущая), или как действительная в ближайшем будущем (следующая). Это позволяет передавать данные о будущей конфигурации до изменения, давая декодеру возможность подготовиться к переменам. Однако, передавать заранее новую версию секции вовсе не обязательно, но, если она передается, то она должна стать следующей правильной версией этой секции.

C.3 Преобразование секций в пакеты Транспортного потока

Секции преобразуются непосредственно в пакеты Транспортного потока, то есть, без предварительного преобразования в пакеты PES. Секции не должны начинаться в начале пакетов Транспортного потока, (хотя они могут там начинаться), потому что начало первой секции полезной нагрузки пакета Транспортного потока указывается полем pointer_field. О наличии поля pointer_field сообщается в пакетах PSI индикатором payload_unit_start_indicator, установленным в значение '1'. (В пакетах, не являющихся пакетами PSI, этот индикатор сообщает, что пакет PES начинается в пакете Транспортного потока.) Поле pointer_field указывает начало первой секции в пакете Транспортного потока. В пакете Транспортного потока никогда не бывает более одного поля pointer_field, поскольку начало любой секции может быть определено путем подсчета продолжительности первой и всех последующих секций, поскольку синтаксис не допускает никаких пробелов между секциями в пакете Транспортного потока.

Важно отметить, что в пакетах Транспортного потока с любым значением PID, одна секция должна быть завершена до того, как может быть начата другая, так как в ином случае невозможно определить, к заголовку какой секции принадлежат данные. Если секция завешается до окончания пакета Транспортного потока, но начинать другую секцию неудобно, используется механизм заполнения, который заполняет пространство. Заполнение выполняется путем введения значения 0xFF в каждый оставшийся байт пакета. Следовательно, значение table_id = 0xFF является запрещенным, так как в ином случае оно может быть принято за заполнение. После появления байта 0xFF в конце секции, весь остаток пакета Транспортного потока должен быть заполнен байтами 0xFF, позволяя декодеру отбросить остаток пакета Транспортного потока. Заполнение может быть выполнено также с использованием обычного механизма adaptation_field.

C.4 Частота повторений и случайный доступ

В системах, где рассматривается случайный доступ, секции PSI рекомендуется передавать несколько раз, даже, если нет изменений конфигурации. Так как в общем случае декодеру требуются данные PSI для идентификации содержания Транспортного потока, с тем, чтобы иметь возможность начать декодирование. В настоящей Рекомендации | Международном стандарте не устанавливается никаких требований относительно частоты повторения или появления секций PSI. Тем не менее, ясно, что частое повторение секций упрощает случайный доступ к приложениям, хотя приводит к увеличению скорости передачи, используемой в данных PSI. Если данные о преобразовании программы статичны или квазистатичны, они могут быть сохранены в декодере, для того чтобы обеспечить более быстрый доступ к данным, так как при этом нет необходимости ждать их повторной передачи. Соотношение между требуемым объемом памяти и желаемым влиянием на время доступа к каналу может быть определено производителем декодера.

C.5 Что такое программа?

Концепция программы точно определена в настоящей Рекомендации | Международном стандарте [см. § 2.1.60 программа (система)]. Для Транспортного потока временная база определяется Эталонным временем программы (PCR). Она эффективно создает в Транспортном потоке виртуальный канал.

Отметим, что это не то же самое определение, которое обычно применяется в радиовещании, где "программа" представляет собой набор элементарных потоков, не только с общей временной базой, но также и с общим временем начала и окончания. Наборы "радиовещательных программ" (которые в настоящем Приложении называются "событиями") могут передаваться в Транспортном потоке последовательно с использованием одного и того же значения program_number, формируя "обычный радиовещательный" ТВ канал (который иногда называется услугой).

Описания событий могут передаваться в секциях private_section().

Программа обозначается номером program_number, который имеет смысл только в Транспортном потоке. Номер program_number – это 16-битовое целое число без знака, и, следовательно, позволяет иметь в Транспортном потоке 65535 уникальных программ (значение program_number "0" зарезервировано для идентификации таблицы NIT). Когда на декодере имеется несколько Транспортных потоков (например, в кабельной сети), для успешного демультиплексирования программы, декодер должен знать для данной услуги как transport_stream_id (для того чтобы найти нужный мультиплексированный поток), так и program_number (для того чтобы найти в мультиплексированном потоке нужную программу).

Преобразование Транспортного потока может быть выполнено при помощи дополнительной Таблицы сетевой информации. Отметим, что для сокращения времени доступа к каналу Таблица сетевой информации может быть сохранена в оперативной памяти декодера. В таком случае, она должна передаваться только так часто, чтобы своевременно поддерживать операции первичной установки декодера. Содержание таблицы NIT является конфиденциальным, но оно должно занимать, как минимум, минимальную структуру секции.

C.6 Распределение значений program_number

Во всех случаях может оказаться удобным сгруппировать вместе в виде одной программы все программные элементы, которые совместно используют общую опорную частоту. Можно представить наличие мультисервисного Транспортного потока, у которого имеется один общий для всех набор значений PCR. Как правило, радиовещатель предпочитает логически разбить Транспортный поток на несколько программ, в которых PCR_PID (местоположение опорной частоты) всегда одинаково. Этот метод разбиения элементов программы на псевдонезависимые программы имеет несколько вариантов использования. Ниже приведено два примера:

i) *Многоязычная передача в различные пакеты*

Один видеопоток может сопровождаться несколькими аудиопотоками на различных языках. Рекомендуется вводить в поток пример дескриптора ISO_639_language_descriptor, связанного с каждым аудиопотоком, для того чтобы иметь возможность выбрать правильную программу и нужный аудиопоток. Целесообразно иметь несколько определений программы с различными значениями program_number, где вес программы указывают одни и те же идентификатор видеопотока и PCR_PID, но имеют различные значения PID для аудиопотока. Однако, также целесообразно и возможно перечислить видеопоток и все аудиопотоки как одну программу, если при этом не будет превышен предельный для этой секции размер 1024 байтов.

ii) *Очень большие определения программы*

Существует максимальный предел для секции, который составляет 1024 байта (включая заголовок секции и CRC_32). Это значит, то ни одно определение программы не может превышать этой длины. Для подавляющего большинства случаев, даже когда каждый элемент программы имеет несколько дескрипторов, этого размера вполне хватает. Однако, можно

предвидеть возможность существования систем с очень большой скоростью передачи, которые могут превышать этот предел. Следовательно, в общем случае, можно определить методы разбиения обозначений потоков, так, чтобы они не должны были перечисляться все вместе. Некоторые элементы программы могут указываться в нескольких программах, а некоторые – только в одной или в другой, но не в обеих вместе.

C.7 Использование PSI в типичной системе

Система связи, особенно в радиовещательных приложениях, может состоять из множества отдельных Транспортных потоков. В каждом из Транспортных потоков системы может появляться любая из четырех структур данных PSI. Всегда должна существовать полная версия Таблицы ассоциаций программы, в которой перечислены все программы, передаваемые в Транспортном потоке, и полная Таблица преобразования программы TS, содержащая полные определения программы для всех программ в Транспортном потоке. Если какие-либо потоки скремблированы, то должна существовать также таблица условного доступа, в которой перечисляются действительные потоки сообщений, предоставляющих право доступа (EMM). Наличие таблицы NIT совсем необязательно.

Таблицы PSI преобразуются в пакеты Транспортного потока при помощи вышеописанной структуры секций. В заголовке каждой секции имеется поле table_id, которое позволяет смешивать секции из таблиц PSI и конфиденциальную информацию в секциях private_sections пакетов Транспортного потока с одним и тем же значением PID или даже в одном пакете Транспортного потока. Отметим, однако, что в пакете с одним и тем же значением PID полная секция должна быть передана до начала следующей секции. Однако, это возможно только для пакетов, отмеченных как содержащие секцию Таблицы преобразования программы TS или пакеты NIT, поскольку конфиденциальные секции не могут быть преобразованы в пакеты PAT или CAT.

Требуется, чтобы все секции таблицы PAT преобразовывались в пакеты Транспортного потока с PID = 0x0000, и все секции CA преобразовывались в пакеты с PID = 0x0001. Секции таблицы PMT могут быть преобразованы в пакеты с выбранным пользователем значением PID, которое указано как PMT_PID для каждой программы в Таблице ассоциаций для программы. Аналогично, значение PID для пакетов Транспортного потока, содержащих NIT, выбирается пользователем, но, если NIT существует, то это должно быть записью "program_number = = 0x00" в таблице PAT.

Содержание любых потоков параметров CA полностью конфиденциально, но сообщения EMM и ECM должны также передаваться в пакетах Транспортного потока, для того чтобы соответствовать настоящей Рекомендации | Международному стандарту.

Таблицы конфиденциальной информации могут передаваться с использованием синтаксиса конфиденциальных секций private_section(). Такие таблицы могут применяться, например, в радиовещании для описания услуги, ближайшего события, расписания вещания и связанной с ними информации.

C.8 Взаимосвязи структур PSI

На рисунке C.1 показан пример взаимосвязи между четырьмя структурами PSI и Транспортным потоком. Возможны и другие примеры, но на рисунке показаны основные связи.

В последующих подразделах описана каждая таблица PSI.

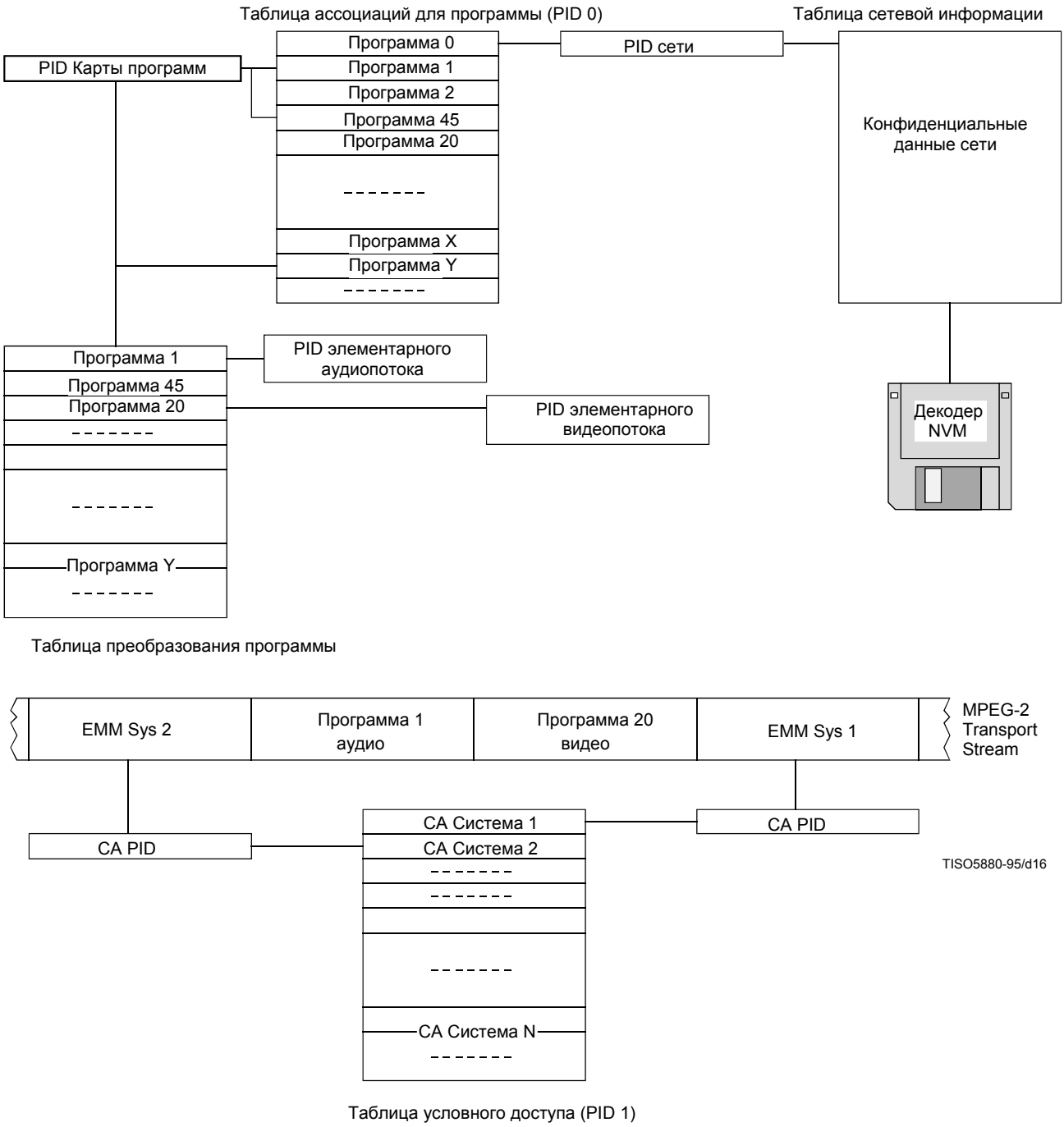


Рисунок C.1 – Взаимосвязь преобразования программы и сети

C.8.1 Таблица ассоциаций для программы

Каждый Транспортный поток должен содержать полную действующую Таблицу ассоциаций для программы (PAT). Таблица ассоциаций для программы определяет соответствие между значением program_number и PID пакетов Транспортного потока, в которых предается описание этой программы (PMT_PID). Перед преобразованием в пакеты Транспортного потока таблица PAT может быть разделена на секции, число которых может достигать 255. В каждой секции передается часть полной таблицы PAT. Такое деление может оказаться желательным для минимизации потерь данных в условиях наличия ошибок. Таким образом, можно локализовать потери пакетов или ошибки в битах в более мелких секциях таблицы PAT, что позволяет продолжать принимать и корректно декодировать остальные секции. Если вся информация PAT помещена в одну секцию, то ошибка, например, приводящая к изменению бита в идентификаторе table_id, вызовет потерю всей таблицы PAT. Однако это допускается до тех пор, пока длительность секции не превышает предела максимальной длины, т. е. 1024 байтов.

Программа 0 (ноль) резервируется и используется для определения PID сети. Это указатель на пакеты Транспортного потока, в который передается Таблица сетевой информации.

Таблица ассоциаций для программы всегда передается без шифрования.

C.8.2 Таблица преобразования программы

Таблица преобразования программы обеспечивает преобразование между номером программы и элементами программы, из которых она состоит. Эта таблица присутствует в пакетах Транспортного потока, имеющих одно или несколько конфиденциально выбранных значений PID. Эти пакеты Транспортного потока могут содержать другие конфиденциальные структуры, которые определены в поле table_id. Допускается иметь секции таблицы PMT TS, ссылающиеся на различные программы, передаваемые в пакетах Транспортного потока, имеющих общее значение PID.

В настоящей Рекомендации | Международном стандарте требуется минимальная идентификация программы: номер программы, PCR PID, типы потоков и значения PID элементов программы. Дополнительная информация о программах или элементарных потоках может передаваться с использованием конструкции descriptor(). См. § C.8.6.

Конфиденциальная информация может передаваться также в пакетах Транспортного потока, которые обозначены, как содержащие секции Таблицы преобразования программы TS. Это осуществляется при помощи секции private_section(). В секции private_section() приложение решает, используются ли значения полей version_number и current_next_indicator для отдельной секции, или они применимы к нескольким секциям, являющимся частями более крупной конфиденциальной таблицы.

ПРИМЕЧАНИЕ 1. – Пакеты Транспортного потока, содержащие Таблицу преобразования программы, передаются в нешифрованном виде.

ПРИМЕЧАНИЕ 2. – Имеется возможность передавать информацию о событиях в конфиденциальных дескрипторах, которые передаются в секциях TS_program_map_section().

C.8.3 Таблица условного доступа

Таблица условного доступа (CA) показывает связь между одной или несколькими системами CA, их потоками EMM и любыми дополнительными параметрами, связанными с ними.

ПРИМЕЧАНИЕ. – (Конфиденциальное) содержание пакетов Транспортного потока, содержащих параметры EMM и CA, если они присутствуют, как правило, будут зашифрованы (скремблированы).

C.8.4 Таблица сетевой информации

Содержание таблицы NIT является конфиденциальным и настоящей Рекомендацией | Международным стандартом не определяется. Как правило, она будет содержать данные о преобразовании выбранных пользователем услуг с идентификаторами transport_stream_id, частоты каналов, номера спутниковых транспондеров, характеристики модуляции и т. п.

C.8.5 Секция private_section()

Секции private_section() могут появляться в двух базовых форматах, коротком (в который включены только поля, расположенные до поля section_length, включительно) и длинном (в который включены все поля, расположенные до поля last_section_number, включительно, а после байтов конфиденциальной информации расположено поле CRC_32).

Секции private_section() могут появляться в идентификаторах PID, которые помечены, как PMT_PID, или в пакетах Транспортного потока с другими значениями PID, которые содержат только секции private_sections(), включая идентификатор PID, распределенный таблице NIT. Если пакеты Транспортного потока с идентификатором PID, в которых передаются секции private_section(), определены как идентификаторы PID, в которых передаются секции private_sections (назначенное значение stream_type = 0x05), то в пакетах Транспортного потока с таким значением PID могут находиться только секции private_section. Это могут быть секции как короткого, так и длинного типа.

C.8.6 Дескрипторы

Существует несколько нормативных дескрипторов, определенных в настоящей Рекомендации | Международном стандарте. Кроме того, может быть определено намного больше конфиденциальных дескрипторов. Все дескрипторы имеют общий формат: {tag, length, data}. Любой дескриптор, определенный в конфиденциальном порядке, должен соответствовать этому формату. Участок "данные" этих конфиденциальных дескрипторов определяется в конфиденциальном порядке.

Один дескриптор (дескриптор условного доступа CA_descriptor()) используется для определения местоположения (значение PID транспортных пакетов) данных сообщений ECM, связанных с элементами программы, когда она находится в секции таблицы PMT Транспортного потока (TS PMT). Когда она находится в секции условного доступа (CA), она ссылается на сообщения EMM.

Для расширения числа доступных дескрипторов private_descriptor, может применяться следующий механизм: Метка private_descriptor_tag может быть определена в конфиденциальном порядке, как созданная в виде композитного дескриптора. Это влечет за собой конфиденциальное определение дальнейшего субдескриптора (sub_descriptor) как первого поля байтов конфиденциальной информации конфиденциального дескриптора. Эта описанная структура имеет вид, показанный в таблицах C.1 и C.2.

Таблица C.1 – Композитный дескриптор (composite_descriptor

Синтаксис	Количество битов	Мнемоника
Composite_descriptor(){		
descriptor_tag (определен в конфиденциальном порядке)	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++){		
sub_descriptor()		
}		
}		

Таблица C.2 –Sub-descriptor

Синтаксис	Количество битов	Мнемоника
sub_descriptor() {		
sub_descriptor_tag	**8**	**uimsbf**
sub_descriptor_length	**8**	**uimsbf**
for (i = 0; i < N; i++) {		
private_data_byte	**8**	**uimsbf**
}		
}		

C.9 Использование полосы пропускания и время обнаружения сигнала

Любой вариант реализации двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, должен формировать запросы на необходимую полосу пропускания для передачи информации PSI и, в тех приложениях, где рассматривается случайный доступ, он должен способствовать быстрому обнаружению сигнала. В настоящем подразделе анализируется эта задача, и приводятся некоторые примеры радиовещательных приложений.

Пакетная природа Транспортного потока позволяет выполнять перемежение информации PSI в мультиплексированном потоке с разбиением на более мелкие участки. Это обеспечивает значительную гибкость при формировании и передаче PSI.

Время обнаружения сигнала в реальном декодере зависит от многих факторов, включая: время настройки частотного мультиплексирования (FDM), время демультиплексирования, заголовки последовательности, частоту появления I-кадров, а также получение ключа скремблирования и обработку.

В настоящем подразделе рассматривается влияние на синтаксис PSI (§ 2.4.4.4 и § 2.4.4.9) как скорости передачи, так и времени обнаружения сигнала. Предполагается, что нет необходимости динамического приема Таблицы условного доступа при каждом изменении программы. Такое же предположение сделано и для конфиденциальных потоков EMM. Причиной этого является то, что эти потоки не содержат быстроменяющихся компонентов ECM, используемых для скремблирования (шифрования) элементов программы.

Кроме того, в приведенных далее рассуждениях не учитывалось время обнаружения и обработки сообщений ECM.

В таблицах C.3 и C.4 представлены данные об использовании полосы пропускания для широкого диапазона условий Транспортного потока. По одной из осей таблицы показано число программ, содержащихся в отдельном Транспортном потоке. По другой – частота, с которой в этом Транспортном потоке передается информация PSI.

Таблица C.3 – Таблица ассоциаций для программы – использование полосы пропускания (бит/с)
Число программ в Транспортном потоке

		1	5	10	32	128
Частота в Таблице ассоциаций для программы Информация (c^{-1})	1	1504	1504	1504	1504	4512
	10	15040	15040	15040	15040	45120
	25	37600	37600	37600	37600	112800
	50	75200	75200	75200	75200	225600
	100	150400	150400	150400	150400	451200
ПРИМЕЧАНИЕ. – Поскольку в **одном транспортном пакете** размещается 46 **секций ассоциаций для программы**, цифры, приведенные в таблице, остаются неизменными до последнего столбца						

Таблица C.4 – Таблица преобразований программы – использование полосы пропускания (бит/с)
Число программ в Транспортном потоке

		1	5	10	32	128
Частота в Таблице преобразований программы Информация (c^{-1})	1	1504	1504	3008	7520	28576
	10	15040	15040	30080	75200	285760
	25	37600	37600	75200	188000	714400
	50	75200	75200	150400	376000	1428800
	100	150400	150400	300800	601600	2857600

Эта частота будет ключевым детерминантом компонента времени обнаружения сигнала, определяемого структурой PSI.

В обеих таблицах использования полосы пропускания предполагается наличие только минимального объема информации о преобразовании программы. Это означает, что значения PID и Типы потоков представлены без каких-либо дополнительных дескрипторов. Все программы в примере составлены из двух элементарных потоков. Ассоциации для программы имеют длину 2 байта, тогда как минимальная карта преобразований программы имеет длину 26 байтов. Имеется дополнительная информация, связанная с номерами версий, длиной секций и т. п. Она будет занимать порядка 1–3% от общего использования скорости передачи PSI в секциях, имеющих длительность от умеренной до максимальной (от нескольких сотен байтов до 1024 байтов) и, следовательно, здесь она в расчет приниматься не будет.

Вышеприведенные предположения позволяют преобразовывать сорок шесть (46) ассоциаций для программы в один пакет Транспортного потока Таблицы ассоциаций для программы (если не представлено ни одного поля адаптации). Аналогично, в один пакет Транспортного потока помещается семь (7) секций TS_program_map_sections. Можно отметить, что для упрощения динамического мультиплексирования, в одном идентификаторе PMT_PID можно передавать только одну (1) секцию TS_program_map_section. Однако, это может привести к нежелательному увеличению использования скорости передачи PSI.

Использование частоты 25 Гц в этих двух таблицах PSI приводит к тому, что они вносят наихудший из возможных вклад в увеличение времени обнаружения сигнала, равный примерно 80 мс. Это может произойти только тогда, когда требуемые данные Таблицы ассоциация для программы (PAT) были "только что пропущены", а, затем, после того, как таблица PAT обнаружена и декодирована, оказалось, что требуемые данные Таблицы преобразования программы (PMT) также "только что пропущены". Такое удвоение наихудшего времени обнаружения является одним из недостатков дополнительного уровня косвенной адресации, который вводится структурой PAT. Это влияние может быть уменьшено путем координированной передачи соответствующих пакетов PAT и PMT. Предположительно, преимуществом, которое такой подход предлагает для операций повторного динамического мультиплексирования, является компенсация недостатков.

При частоте PSI, равной 25 Гц, могут быть сформированы следующие примеры (во всех примерах оставлены значительные допуски для различных заголовков линий передачи данных, FEC, CA и маршрутизации):

Канал кабельного ТВ шириной 6 МГц

- пять программ по 5,2 Мбит/с: 26,5 Мбит/с (включая транспортный заголовок)
- общая ширина полосы PSI: 5,2 кбит/с
- ширина полосы CA: 500 кбит/с

  общая ширина полосы транспортировки, соответствующая Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1: 27,1 Мбит/с
- дополнительная информация PSI: 0,28%

Канал в оптоволокне OC-3 (155 Мбит/с)

- 32 программы по 3,9 Мбит/с: 127,5 Мбит/с (включая транспортный заголовок)
- общая ширина полосы PSI: 225,6 кбит/с
- ширина полосы CA: 500 кбит/с

  общая ширина полосы транспортировки, соответствующая Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1: 128,2 Мбит/с
- дополнительная информация PSI: 0,18%

Спутниковый транспондер C-диапазона

- 128 аудиопрограмм по 256-кбит/с: 33.5 Мбит/с (включая транспортный заголовок)
- общая ширина полосы PSI: 826.4 кбит/с
- ширина полосы CA: 500 кбит/с

  общая ширина полосы транспортировки, соответствующая Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1: 34,7 Мбит/с
- дополнительная информация PSI: 2,4% (в действительности, она будет меньше, если для отдельной программы используется только дин PID)

Как ожидается, для услуг с уменьшением скорости передачи процент дополнительной информации возрастает, поскольку в одном Транспортном потоке можно передавать намного больше таких услуг. Однако, дополнительная информация избыточна не во всех случаях. Более высокие скорости передачи данных PSI (более 25 Гц) могут использоваться для снижения влияния дополнительной информации на время обнаружения канала только, если повышается спрос на умеренные скорости передачи.

Приложение D

Модель синхронизации систем и применение положений настоящей Рекомендации | Международного стандарта

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

D.0 Введение

Спецификация систем в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 содержит определенную модель синхронизации для дискретизации, кодирования, буферизации в кодере, передачи, приема, буферизации в декодере, декодирования и воспроизведения комбинации цифровых аудиовизуальных сигналов. Эта модель вводится непосредственно в спецификацию синтаксических и семантических требований для потоков данных, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Учитывая, что система декодирования принимает двоичный поток, который доставлен корректно в соответствии с моделью синхронизации, достаточно просто создать такой декодер, на выходе которого высококачественные аудио и видео сигналы оказываются соответствующим образом синхронизированы. Однако, нет нормативных требований, согласно которым декодеры должны быть построены таким образом, чтобы обеспечивать такой качественный выходной сигнал воспроизведения. В тех приложениях, где данные не доставляются на декодер с корректной синхронизацией, может иметь возможность создать желаемый выходной сигнал воспроизведения; однако, как правило, такие возможности не гарантируются. В настоящем информативном Приложении подробно описывается модель синхронизации систем, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, и приводятся некоторые предложения по реализации декодера системы, для того чтобы он соответствовал некоторым типичным приложениям.

D.0.1 Модель синхронизации

Система, соответствующая Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, включает в себя модель синхронизации, в которой все цифровые изображения и отсчеты звуковых сигналов, которые поступают в кодер и появляются на выходе декодера, воспроизводятся только один раз после определенной постоянной сквозной задержки передачи. Таким образом, значения частоты дискретизации, т. е. частота видеокадров и частота отсчетов звукового сигнала на декодере точно такие, какими они были на кодере. Диаграмма этой модели синхронизации показана на рисунке D.1:

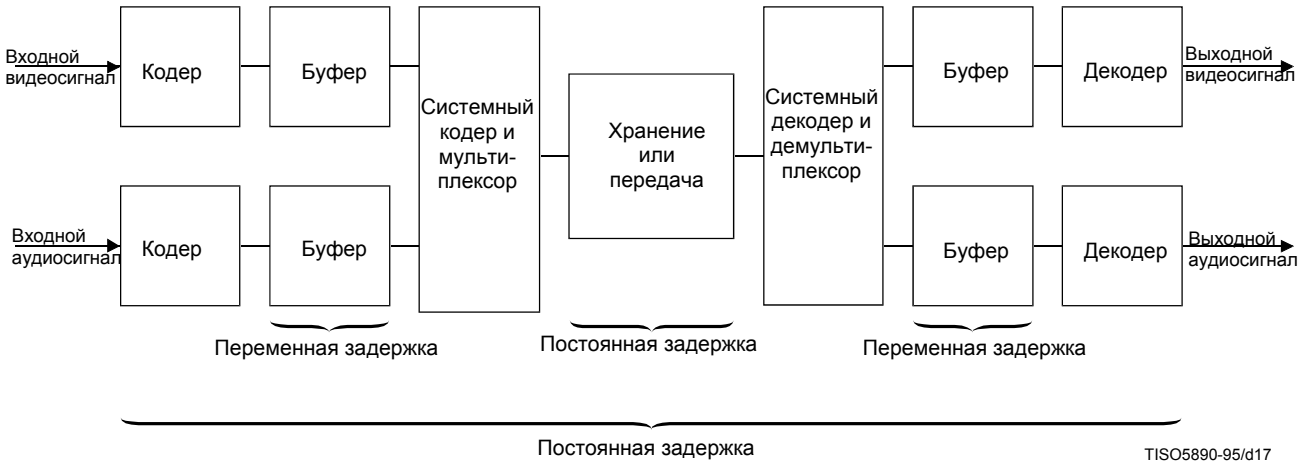


Рисунок D.1 – Модель с постоянной задержкой

Как показано на рисунке D.1, в данной модели задержка от входа кодера до выхода декодера или воспроизведения постоянна[1)], тогда как задержка в буфере каждого кодера и декодера переменна. Переменной является не только задержка в каждом из этих буферов на пути одного элементарного потока, различаются также и задержки в отдельных буферах для аудио и видео сигналов. Следовательно, относительное местоположение в комбинированном потоке кодированных битов, описывающих звук и изображение, не указывает информацию синхронизации. Относительное местоположение кодированных аудио и видео сигналов ограничивается моделью декодера конечной системы (STD) только таким образом, что буферы декодеров должны функционировать определенным порядком; следовательно, кодированные аудио и видео сигналы, представляющие собой звук и изображение, которые должны воспроизводиться одновременно, в пределах кодированного двоичного потока могут отстоять друг от друга на время до одной секунды, что является максимальным значением задержки буфера декодера, допускаемой в модели STD.

Значения частоты синхронизации аудио и видео сигналов в кодере значительно отличаются друг от друга, и могут быть связаны или не связаны друг с другом, в зависимости от того, является ли комбинированный поток Программным или Транспортным, и от того, установлены ли в Программном потоке флаги System_audio_locked и System_video_locked. Продолжительность блоков отсчетов аудиосигнала (блок представления аудиосигнала), как правило, не равна продолжительности видеокадра.

В кодере имеются одни общие системные часы, и они используются для формирования меток времени, которые обозначают корректное время воспроизведения и декодирования для звука и изображения, а также для формирования меток времени, которые обозначают мгновенные значения самих системных часов на интервалах дискретизации. Метки времени, которые указывают время воспроизведения звука и изображения, называются Метками времени воспроизведения (PTS). Метки времени, которые указывают время декодирования, называются Метками времени декодирования (DTS), а метки времени, которые указывают значения системных часов, называются Эталонным временем системы (SCR) в Программных потоках и Эталонным временем программы (PCR) в Транспортных потоках. Именно наличие этих общих системных часов в кодере, метки времени, которые ими создаются и восстанавливают время в декодере, а также правильное использование меток времени дают возможность правильно синхронизировать работу декодера.

Варианты реализации кодера могут не соответствовать этой модели в точности; однако, поток данных, который создается на выходе реального кодера, системы хранения, сети и одного или нескольких мультиплексоров, должен соответствовать этой модели абсолютно точно. (Доставка данных может несколько отличаться в зависимости от приложения.) Следовательно, в настоящем Приложении термин "системные часы кодера" используется для обозначения либо реальных общих системных часов, которые описаны в настоящей модели, либо эквивалентной им функции, как бы она ни была реализована.

Поскольку сквозная задержка передачи через всю систему постоянна, процессы воспроизведения аудио и видеосигналов точно синхронизированы. Формирование системных двоичных потоков ограничено таким образом, что когда они декодируются декодером, который соответствует этой модели и имеет правильно подобранные размеры буферов декодера, гарантируется, что эти буферы никогда не будут ни переполняться, ни оставаться недозаполненными, за исключением особых случаев, когда недозаполнение осуществляется намеренно.

---

1) Постоянная задержка, указанная для всей системы в целом, требуется для корректной синхронизации, однако, возможны некоторые отклонения. Задержка в сети считается постоянной, однако, могут допускаться небольшие отклонения, и адаптация сети может допускать более заметные изменения задержки в сети. Оба эти варианта рассматриваются далее.

Для того чтобы система декодера вносила задержку точной продолжительности, которая приводит к получению постоянной сквозной задержки, необходимо, чтобы декодер имел системные часы, частота работы которых и абсолютное мгновенное значение которых совпадали бы с этими же значениями на кодере. Информация, необходимая для передачи данных системных часов кодера, кодируется в полях эталонного времени системы (SCR) или эталонного времени программ (PCR); эта функция рассматривается далее.

Декодеры, которые реализованы в соответствии с этой моделью синхронизации, таким образом, что они представляют отсчеты аудиосигнала и видеокадры только один раз (за исключением особых случаев, в которых различие в воспроизведении специально кодируются), с постоянной скоростью, и так, что буферы декодера функционируют в соответствии с моделью, в настоящем Приложении называются точно синхронизированными декодерами, или декодерами, которые формируют точно синхронизированный выходной сигнал. В настоящем Международном стандарте не требуется, чтобы варианты реализации декодера представляли аудио и видеосигналы в соответствии с этой моделью; можно создать декодеры, которые не имеют постоянной задержки, или, что равнозначно, не обеспечивают точно одновременного воспроизведения каждого кадра изображения и отсчета аудиосигнала. Однако, в таких вариантах реализации синхронизация между воспроизводимыми аудио и видеосигналами может оказаться не точной, и функционирование буферов декодера может не соответствовать эталонной модели декодера. Очень важно предотвратить переполнение буферов декодера, поскольку их переполнение приводит к потере данных, которая может существенно повлиять на результат процесса декодирования. В настоящем Приложении рассматривается, главным образом, работа точно синхронизированных декодеров и некоторые существующие возможности для реализации таких декодеров.

D.0.2 Синхронизация воспроизведения аудио и видеосигналов

В системах кодирования, соответствующих настоящей Рекомендации | Международному стандарту, данными являются метки времени, относящиеся к воспроизведению и декодированию видеокадров и блоков отсчетов аудиосигнала. Эти видеокадры и блоки называются "Модулями представления", сокращенно – PU. Множество кодированных битов, которые представляют собой модули PU, и которые находятся в составе двоичного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, называются "Модулями доступа", сокращенно – AU. Модуль доступа аудиосигнала сокращенно обозначается AAU, а модуль доступа видеосигнала сокращенно обозначается VAU. В стандарте ИСО/МЭК 13818-3 "звук" термин "звуковой кадр" имеет такое же значение, что и AAU или APU (модуль представления аудиосигнала) в зависимости от контекста. Модуль представления видеосигнала (VPU) – это кадр изображения, а VAU – это кодированный кадр.

Некоторые, но не обязательно все, модули AAU и VAU имеют связанные с ними метки времени PTS. Метка PTS указывает момент времени, в который PU, полученный в результате декодирования модуля AU, который связан с этой PTS, должен быть представлен пользователю. Метки времени PTS аудио и видео сигналов являются значениями общего задающего генератора, который называется системными часами или STC. Точная синхронизация представляемых аудио и видеосигналов обеспечивается в системе декодирования, когда в поток данных введены корректные метки времени PTS аудио и видео сигналов, и когда воспроизведение модулей PU аудио и видео сигналов выполняется в моменты времени, указанные соответствующими метками PTS в единицах общих системных часов STC. Несмотря на то что STC не входит в нормативное содержание настоящей Рекомендации | Международного стандарта, и соответствующая информация вводится в настоящую Рекомендацию | Международный стандарт при помощи таких терминов, как system_clock_frequency (частота системных часов), STC является важной частью и удобным элементом для объяснения модели синхронизации, и, как правило, варианты практической реализация кодеров и декодеров будут включать в свой состав STC в какой-либо форме.

Значения меток PTS требуются для передачи точной информации об относительной синхронизации между аудио и видеосигналами, поскольку модули PU аудио и видеосигналов, как правило, имеют различную и не связанную друг с другом продолжительность. Например, модули PU аудиосигналов, равные 1152 отсчетам, следующих с частотой 44 100 отсчетов в секунду, имеют длину примерно 26,12 мс, а модули PU видеосигналов с частотой кадров 29,97 Гц имеют длину примерно 33,76 мс. Как правило, временные границы модулей APU и VPU редко совпадают (если совпадают вообще). Отдельные метки PTS для аудио и видеосигналов предоставляют информацию, которая содержит точное временное соотношение модулей PU аудио и видеосигналов, не налагая никаких особых требований на соотношение между длительностью модулей PU аудио и видеосигналов и интервалами между ними.

Значения полей меток PTS определяются для декодера конечной системы (STD), который является основным нормативным ограничением для всех двоичных потоков системы. Декодер STD представляет собой математическую модель идеального декодера, который в точности определяет движение всех битов по направлению в буферы декодера и из них, и основное семантическое ограничение, накладываемое на двоичный поток, заключается в том, что буферы в декодере STD никогда не должны ни переполняться, ни оставаться недозаполненными, с определенными исключениями, предусмотренными для особых случаев недозаполнения. В теоретической модели STD декодер всегда точно синхронизирован с источником данных, а также точно синхронизированы декодирование и воспроизведение аудио и видеосигналов.

Предположение о точности и полном соответствии несколько упрощает STD по сравнению с физической реализацией декодеров, его цель сделать спецификацию STD более понятной и содействовать его широкому применению в различных вариантах реализации декодера. В частности, в модели STD каждая операция, которая выполняется в декодере над двоичным потоком, выполняется мгновенно и очевидным исключением из рассмотрения того времени, в течение которого биты находятся в буферах декодеров. В реальной системе декодера отдельные декодеры аудио и видеосигналов не работают мгновенно, и при разработке варианта реализации следует учитывать вносимые ими задержки. Например, если изображения декодируются точно через интервал времени, равный одному кадру (1/P), где P – частота кадров, и компрессированные видеоданные поступают в декодер со скоростью R, то завершение удаления битов, связанных с этим изображением, задерживается относительно времени, указанного в полях меток PTS и DTS на время 1/P, и буфер декодера видеосигнала должен быть на R/P больше, чем указано в модели STD. Воспроизведение видеосигнала точно также задерживается относительно STD, и метка PTS должна быть соответствующим образом обработана. Поскольку видеосигнал задерживается, то декодирование и воспроизведение аудиосигнала также должны быть задержаны на аналогичные интервалы времени, для того чтобы обеспечивалась правильная синхронизация. Задержка декодирования и воспроизведения аудио и видеосигналов в декодере может быть реализована, например, путем добавления постоянных значений к меткам PTS, когда они используются в декодере.

Еще одно различие между STD и точным практическим вариантом реализации декодера заключается в том, что в модели STD делается явное предположение о том, что окончательные выходные аудио и видеосигналы представляются пользователю мгновенно без дополнительных задержек. На практике это совсем не так, особенно в дисплеях с электронно-лучевой трубкой, и эта дополнительная задержка также должна учитываться при разработке. Требуется, чтобы кодеры кодировали аудио и видеосигналы таким образом, чтобы при декодировании данных в STD обеспечивалась точная синхронизация. В кодере должны быть учтены задержки, вносимые на этапах ввода и дискретизации аудио и видеосигналов, например, интеграция оптического изображения видеокамеры.

В модели STD предполагается, что реализована точная синхронизация, а метки времени и функционирование буфера проверяются с учетом этого предположения для условий достоверности двоичного потока. Несомненно, в физической реализации декодера точная синхронизация не всегда осуществима, особенно в режиме запуска и при наличии нестабильности синхронизации. Точная синхронизация декодера является целью, к которой необходимо стремиться при разработке декодеров. Неточность синхронизации декодера влияет на работу буферов декодера. Эти проблемы подробно рассматриваются в последующих подразделах настоящего Приложения.

В состав STD включены поля меток времени DTS и PTS. Метка DTS указывает момент времени, когда модуль доступа AU должен быть извлечен из буфера декодера и декодирован в модели STD. Поскольку декодеры элементарного потока аудио и видеосигналов в модели STD работают мгновенно, время декодирования и время воспроизведения в большинстве случаев идентичны; единственное исключение возникает, когда видеокадры передаются в кодированном двоичном потоке с нарушением исходного порядка следования, т. е. I- кадры и P-кадры для случая видеопоследовательностей не с низкой задержкой. В тех случаях, когда имеется изменение порядка, используется буфер временной задержки для хранения декодированных I- или P-кадров до момента их воспроизведения. Во всех случаях, когда время декодирования и воспроизведения в STD идентичны, т. е. для всех модулей доступа аудиосигнала (AAU), модулей доступа видеосигнала (VAU) B-кадров и модулей VAU I- и P- кадров в видеопоследовательностях с малой задержкой, метка DTS не кодируется, так как ее значение будет точно таким же, как и значение PTS. Когда эти значения отличаются, они кодируются, если закодировано каждое из них. Для всех модулей AU, если кодируется только PTS, то это поле можно интерпретировать, как поле, представляющее собой и PTS, и DTS.

Поскольку значения меток PTS и DTS требуются не для каждого модуля AAU и VAU, декодер может интерполировать значения, которые не были закодированы. Значения меток PTS в каждом элементарном потоке аудио и видеосигналов необходимо иметь с интервалами, не превышающими 700 мс. Эти интервалы времени измеряются для времени воспроизведения, то есть, как некоторый контекст значения полей, а не в единицах времени, когда эти поля передаются и принимаются. В случаях потоков данных, когда системные часы, а также задающие генераторы видео и аудио сигналов синхронизированы, как определено в нормативной части настоящей Рекомендации | Международного стандарта, каждому модулю AU, следующему после модуля AU, для которого метка DTS или PTS закодирована явным образом, соответствует эффективное время декодирования, равное сумме этого времени для предыдущего модуля AU плюс фиксированная и определенная в спецификации разница значения STC. Например, в видеосигнале, кодированном с частотой 29.97 Гц, каждый кадр отличается от предыдущего на время, равное 3003 циклам STC-генератора частоты 90 кГц, если задающие генераторы видео и аудио сигналов синхронизированы. Такое же соотношение присутствует для декодирования последовательных модулей AU, хотя задержка в декодере, обусловленная изменением порядка следования, влияет на соотношение между модулями доступа AU декодера и модулями представления PU. Когда поток данных кодируется таким образом, что задающие генераторы видео и аудио сигналов не синхронизированы с системными часами, разница во времени между декодированием последовательных модулей AU может быть оценена с использованием тех же самых значений, что указано выше; однако, эти значения разницы во времени не будут точными, благодаря тому факту, что на декодере не было определено точного соотношения между частотой кадров, частотой отсчетов аудиосигналов и частотой системных часов.

Отметим, что поля PTS и DTS сами по себе не указывают точной степени заполнения буферов декодера ни в момент запуска, ни в какое-либо другое время, и точно также они не указывает величину задержки во времени, которая должна завершиться в момент приема первых битов потока данных до начала декодирования потока данных. Эта информация определяется в ходе объединения функций полей PTS и DTS и точного восстановления синхронизации, которое описано ниже. В модели STD и, следовательно, в декодерах, которые моделируются в соответствии с ней, работа буфера декодера полностью определяется значениями SCR (или PCR), моментами времени, в которые они принимаются, а также значениями PTS и DTS, в предположении, что эти данные доставляются в соответствии с моделью синхронизации. Эта информация определяет время, в течение которого кодированные данные находятся в буферах декодера. Объем информации, которая находится в буферах кодированных данных, явным образом не определяется, и эта информация не является необходимой, поскольку синхронизация определена полностью в спецификации. Отметим также, что степень заполнения буферов данных может заметно меняться во времени таким образом, который не может быть предсказан в декодере, за исключением случая правильного использования меток времени.

Для того чтобы метки времени PTS аудио и видеосигналов правильно ссылались на общие системные часы STC, в системе декодера должен присутствовать четко синхронизированный общий задающий генератор.

D.0.3 Восстановление времени системных часов в декодере

В системном потоке данных, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, в дополнение к полям PTS и DTS, присутствуют метки времени эталонного времени. Эти эталонные метки представляют собой отсчеты времени системных часов, которые применимы как к декодеру, так и к кодеру. Они имеют разрешение один к 27 000 000 в секунду, и появляются с интервалами до 100 мс в Транспортных потоках, и до 700 мс в Программных потоках. Таким образом, они могут использоваться в декодерах в цепях управления восстановлением синхронизации с точностью, достаточной для всех определенных приложений.

В Программном потоке поле эталонного времени называется эталонным временем системы – SCR. В Транспортном потоке поле эталонного времени называется эталонным временем программы – PCR. Как правило, определения SCR и PCR можно считать эквивалентными, хотя между ними есть различия. В оставшейся части настоящего подраздела для ясности используется термин SCR; те же самые утверждения применимы и для PCR, за исключением случаев, когда сказано обратное. Время PCR в Транспортных потоках обеспечивает данные эталонного времени для одной программы, где программа представляет собой набор элементарных потоков, которые имеют общую временную и предназначены для синхронного декодирования и воспроизведения. В одном Транспортном потоке может существовать несколько программ, каждая из которых может иметь независимую временную базу и отдельный набор значений PCR.

Поле SCR указывает точное значение времени STC, в которое SCR принимается на декодере. Поскольку SCR занимает несколько байтов данных, а системные потоки данных определяются как потоки байтов, SCR определяется, как момент времени прибытия на декодер, в который на декодере принят последний байт поля system_clock_reference_base. Иначе, SCR можно понимать, как момент времени, когда поле SCR должно поступить на декодер, в предположении, что заранее известно, что значение STC правильное. Какую из интерпретаций использовать, зависит от структуры системы приложения. В приложениях, где декодер может управлять источником данных, например, локально соединенный DSM, декодер может иметь автономную частоту STC, и, поэтому, нет необходимости восстанавливать STC. Однако, во многих важнейших приложениях, это предположение неверно. Например, рассмотрим случай, когда поток данных одновременно доставляется на множество декодеров. Если каждый декодер имеет собственную автономную частоту STC со своей собственной независимой тактовой частотой, то невозможно гарантировать доставку значений SCR на все декодеры в нужное время; как правило, на одном декодере значения SCR будут требоваться раньше, чем источник может их доставить, тогда как на другом декодере эти значения потребуются позже. Это различие не может быть скомпенсировано с помощью буфера данных конечного размера для неограниченной продолжительности времени приема данных. Следовательно, далее рассматривается, главным образом, случай, в котором STC должен подстраивать свою синхронизацию в соответствии с принятыми значениями SCR (или PCR).

В правильно созданных и доставленных потоках данных, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, каждое поле SCR поступает в декодер точно в момент времени, указанный значением этого SCR. В этом контексте, "время" означает точное значение STC. По своему характеру, это значение STC является тем же значением, которое имелось в STC кодера, когда сохранялось и передавалось значение эталонного времени системы (SCR). Однако, кодирование могло быть выполнено не в реальном времени, либо поток данных мог быть изменен после своего первоначального кодирования, и, как правило, кодер или источник могли быть реализованы различными способами, в результате значение STC кодера может оказаться теоретическим значением.

Если тактовая частота декодера точно соответствует тактовой частоте кодера, тогда декодирование и воспроизведение видео и аудиосигналов будет автоматически выполняться с той же скоростью, что и в кодере, и сквозная задержка будет постоянной. При точном соответствии таковых частот кодера и декодера для установки STC декодера может использоваться любое корректное значение SCR, и с этого момента времен значение STC декодера будет совпадать с STC кодера без необходимости дальнейшей регулировки. Это

условие продолжает действовать до тех пор, пока не нарушится непрерывность синхронизации, это может быть, при завершении Программного потока или появление индикатора нарушения непрерывности в Транспортном потоке.

На практике автономная системная тактовая частота декодера не будет совпадать тактовой частотой кодера, которая дискретизируется и указывается в значениях SCR. Значения STC декодера могут синхронизироваться от синхронизации кодера, используя принятые значения SCR. Метод-прототип синхронизации тактовой частоты декодера от принятого потока данных выполняется посредством цепи фазовой автоподстройки частоты (ФАПЧ). Могут быть приемлемыми различные варианты базовой схемы ФАПЧ, или иные методы, в зависимости от требований конкретного приложения.

Здесь описана и изображена наиболее простая ФАПЧ, которая восстанавливает значения STC в декодере.

На рисунке D.2 показана классическая ФАПЧ, за исключением того, что эталонными значениями и обратной связью вместо таких событий сигнала, как его окончания, являются численные значения (значения STC и SCR или PCR).

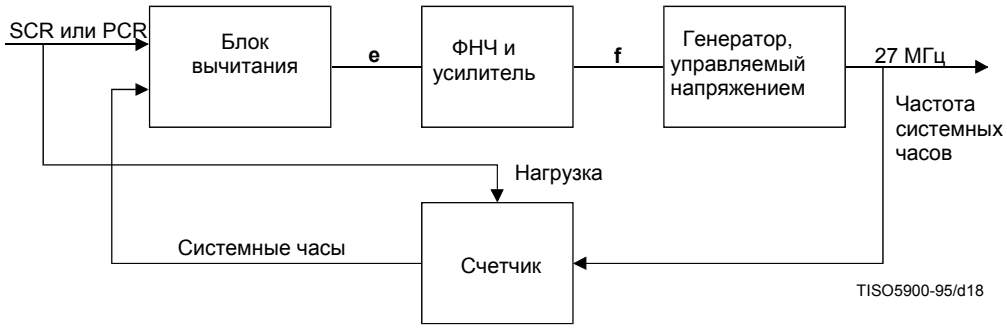


Рисунок D.2 – Восстановление STC с использованием фазовой автоподстройки частоты (ФАПЧ)

После первоначального определения новой временной базы, т. е. новой программы, STC устанавливается в текущее значение, закодированное в полях SCR. Как правило, первое значение SCR загружается непосредственно в счетчик STC, и ФАПЧ, следовательно, работает в виде замкнутой цепи. Могут быть приемлемы и различные вариации этого метода, т. е. если значения SCR не вызывают доверия из-за нестабильности или ошибок.

Работа ФАПЧ в виде замкнутой цепи выполняется следующим образом. В момент времени, когда каждый SCR (или PCR) поступает в декодер, его значение сравнивается с текущим значением STC. Полученная разница представляет собой число, одна часть которого выражена в единицах по 90 кГц, а другая – в единицах в 300 раз превышающих эту частоту, т. е. 27 МГц. Значение разницы линеаризуется и в результате получается продолжительность, выраженная одним числом, обычно, в единицах по 27 МГц, оно называется остаточной ошибкой цепи "**e**". Последовательность значений **e** является входным сигналом для каскада, состоящего из фильтра нижних частот и усилителя, который разработан в соответствии с требования конкретного приложения. Выходной сигнал этого каскада представляет собой сигнал управления "**f**", который управляет мгновенным значением частоты Генератора, управляемого напряжением (ГУН). Выходной сигнал ГУН – это сигнал генератора с номинальной частотой 27 МГц; этот сигнал используется в декодере в качестве сигнала системного тактового генератора. Сигнал с частотой 27 МГц является входным сигналом счетчика, который создает текущие значения STC, которые состоят как из расширения с частотой 27 МГц, полученного в результате деления на 300, так и из базового значения с частотой 9 кГц, которое получается в результате подсчета результатов с частотой 90 кГц в 33-битовом счетчике. 33-юитовая 90 килогерцовая часть выходного сигнала STC используется, при необходимости, для сравнения со значениями PTS и DTS. Полное значение STC также по цепи обратной связи подается на вход блока вычитания.

Ограниченный максимальный интервал между последовательными значениями SCR (700 мс) или PCR (100 мс) позволяет разработать и создать системы ФАПЧ, которые известны как стабильные. Полоса пропускания таких систем ФАПЧ имеет верхнюю границу, определенную этим интервалом. Как показано ниже, во многих приложениях требуемая ФАПЧ имеет очень малую полосу пропускания, и, таким образом, эта граница, как правило. Е накладывает заметных ограничений на разработку и параметры декодера.

Если автономная или исходная частота ГУН достаточно точно совпадает с правильной системной тактовой частотой кодера, то декодер может иметь возможность работать удовлетворительно, до тех пор, пока STC создается корректным образом, прежде, чем ФАПЧ достигнет определенного фиксированного состояния. Для данного декодера, частота STC которого на ограниченную величину отличается от частоты, закодированной в значениях SCR, и которая лежит в абсолютных пределах частоты, требуемых для приложения декодера, результат несоответствия между частотами STC кодера и декодера заключается в том, что, если бы ФАПЧ не использовалась, то имело бы местно постепенное и неизбежное увеличение или уменьшение степени

наполнения буферов декодера, так что в конце концов, случилось бы переполнение иди исчерпывание для любого ограниченного размера буферов декодера. Следовательно, продолжительность времени, допустимого до установления синхронизации частоты STC декодер с этой частотой кодера определяется величиной допустимого дополнительного размера буфера декодера и задержкой.

Если декодером принимаются SCR со значениями и синхронизацией, которые отражают корректные мгновенные отсчеты в кодере с постоянной частотой STC, тогда после того, как цепь достигнет синхронного состояния, ошибка **e** становится практически постоянной. Это условие наличия корректных значений SCR означает либо хранение и передачу данных с кодера на декодер с постоянной задержкой, либо, если эта задержка не постоянна, является эффективным эквивалентом постоянной задержки хранения и передачи со значениями SCR, которые корректируются, для того чтобы отразить изменения величины задержки. Поскольку значения **e** превращаются в постоянные, и после того, как цепь синхронизирована, изменения мгновенной частоты ГУН становятся, практически равными нулю; говорят, что ГУН обладает очень малой нестабильностью или малым уходом частоты. В то время, как цепь находится в процессе синхронизации, скорость изменения частоты ГУН, и скорость спадания функции, могут жестко контролироваться в каскаде фильтра нижних частот и усиления. Как правило, в ГУН можно обеспечить такую скорость спадания сигнала, чтобы выполнялись требования приложения, в зависимости от размера буфера декодера и величины задержки.

D.0.4 Нестабильность SCR и PCR

Если при доставке потока данных от кодера до системы хранения в декодере сеть или ремультиплексор Транспортного потока изменяет вносимую задержку, то такие изменения могут привести к появлению различий между значениями SCR (или PCR) и значениями, которые они должны иметь, когда они принимаются в действительности. Это называется нестабильностью эталонного времени системы (SCR) или программы (PCR). Например, если задержка доставки одного поля SCR больше, чем задержка, испытываемая другими аналогичными полями той же программы, говорят, что это поле SCR отстает. Аналогично, если эта задержка меньше, чем задержка для других полей эталонного времени той же программы, говорят, что это поле появляется слишком рано.

Нестабильность синхронизации на входе декодера отражается в комбинации значений SCR и моментов времени, когда эти значения были приняты. Если предположить, что используется схема восстановления синхронизации, показанная на рисунке D.2, то любая такая нестабильность синхронизации будет описана в значениях ошибки **e**; ненулевые значения **e** приводят к изменениям значений **f**, результатом чего является изменение частоты системных часов (27 МГц). Изменения восстановленной тактовой частоты могут быть приемлемыми или неприемлемыми для декодера системы, в зависимости от требований конкретного приложения. Например, в точно синхронизированных декодерах, которые формируют композитный выходной видеосигнал, и где восстановленная тактовая частота, как правило, используется для создания тактовой частоты отсчета композитного видеосигнала и цветовой поднесущей; применимые спецификации для стабильности поднесущей частоты могут допускать только очень медленную регулировку частоты системных часов. В приложениях, где на входе декодера наблюдается существенная нестабильность SCR или PCR, и применяются строгие ограничения скорости изменения частоты системных часов (STC), установление разумных ограничений размера дополнительного буфера декодера и задержки для обеспечения правильной работы может быть недостаточным.

Появление нестабильности эталонного времени системы (SCR) или программы (PCR) может быть вызвано, например, передачей в сети, при которой используется мультиплексирование пакетов или блоков, или которая вносит различную задержку при передаче пакетов по сети, что может быть вызвано суммированием задержек или различными значениями времени доступа к сети в системах с совместно используемой средой передачи.

Мультиплексирование или ремультиплексирование Транспортных или Программных потоков изменяет порядок и относительные значения времени следования пакетов данных, и, следовательно, полей SCR или PCR. Изменение моментов времени появления полей SCR приводит к тому, что корректные ранее значения SCR становятся неверными, поскольку, как правило, их значения неточно указывают моменты времени, в которые они доставляются по сети с постоянной задержкой. Аналогично, Программный поток или Транспортный поток с правильными значениями SCR или PCR может быть доставлен по сети, которая вносит переменную задержку потока данных, не изменяя при этом значений SCR или PCR. Результатом этого также является нестабильность SCR или PCR, оказывающая соответствующее влияние на структуру и качественные показатели декодера. Наихудшая нестабильность, вносимая сетью в значения принимаемых на декодере полей SCR или PCR, зависит от количества факторов, которые выходят за рамки настоящей Рекомендации | Международного стандарта, включая размеры очередей, реализованные в каждом из коммутаторов сети, и общее число коммутаторов сети или операций ремультиплексирования, через которые последовательно проходит поток данных.

В случае Транспортного потока необходима коррекция значений PCR при выполнении операции ремультиплексирования, в результате которой из одного или нескольких Транспортных потоков создается новый Транспортный поток. Эта коррекция выполняется путем добавления к значениям PCR корректирующего коэффициента; этот коэффициент может быть вычислен следующим образом:

$$\Delta PCR = del_{act} - del_{const},$$

где del_{act} – реальная задержка, испытываемая значением PCR, и del_{const} – постоянная задержка, которая используется для всех значений PCR этой программы. Значение, которое должно использоваться для del_{const}, будет зависеть от стратегии, применяемой в исходном кодере/мультиплексоре. Эта стратегия может, например, заключаться в том, чтобы устанавливать максимально раннее время передачи пакета, для того чтобы дать возможность задержки для тех линий связи, по которым эти пакеты будут передаваться позже. В таблице D.1 показаны три различные стратегии мультиплексирования, а также соответствующие им значения del_{const}.

Таблица D.1 – Стратегия ремультиплексирования

Стратегия	del_{const}
Ранняя передача	del_{min}
Поздняя передача	del_{max}
Средняя по времени передача	de_{lavg}

При разработке системы могут потребоваться конфиденциальные договоренности относительно того, какая стратегия должна использоваться в кодерах/мультиплексорах, поскольку она будет влиять на способность выполнять любое дополнительное ремультиплексирование.

Величина допустимой нестабильности, вносимой мультиплексированием, не ограничивается нормативными требованиями, приведенными в настоящей Рекомендации | Международном стандарте. Однако предполагается, что для работоспособной системы максимальной величиной нестабильности является значение 4 мс.

В системах с ремультиплексорами может потребоваться уделить особое внимание обеспечению согласованности информации в Транспортном потоке. В частности, это относится к PSI и точкам нарушения непрерывности. Изменения в таблицах PSI может потребоваться вводить в Транспортный поток таким образом, чтобы последующие операции ремультиплексирования никогда бы не сместили их настолько, чтобы эта информация стала бы неверной. Например, в некоторых случаях новая версия таблицы PMT не должна передаваться в течение 4 мс после данных, на которые повлияют вносимые изменения.

Аналогично, может потребоваться, чтобы в течение временного окна ±4 мс вокруг точки нарушения непрерывности кодер/мультиплексор не вводил меток времени PTS или DTS.

D.0.5 Восстановление времени системных часов при наличии нестабильности сети

В приложениях, где наблюдается значительная величина нестабильности принятых меток эталонного времени, может быть выбрано несколько вариантов структуры декодера; то, как выполняется декодер, в значительной степени зависит от требований к характеристикам выходного сигнала декодера, а также от характеристик данных на его входе и от нестабильности задержки.

В различных приложениях декодеры могут иметь различные требования к точности и стабильности восстановленного времени системных часов, и можно считать, что требуемая степень этой стабильности и точности расположена вдоль некоторой оси координат. Одна из крайних точек этой оси может приходиться на те приложения, в которых восстановленные значения времени системных часов используются непосредственно для синхронизации цветовой поднесущей в композитном видеосигнале. Это требование, обычно, накладывается, когда воспроизводимый видеосигнал имеет вышеописанный тип с точной синхронизацией, в котором каждый кодированный видеокадр представляется только один раз, и в котором выходной сигнал является композитным видеосигналом, соответствующим применяемым спецификациям. В таком случае цветовая поднесущая, частота следования пикселов и частота кадров имеют точно определенные значения, и все они определенным образом связаны с частотой системных часов. Поднесущая композитного видеосигнала должна иметь точность и стабильность, достаточную, как минимум, для того чтобы в любом обычном телевизионном приемнике от этой поднесущей могла бы синхронизироваться ФАПЧ цветовой поднесущей, и чтобы в сигналах цветности, которые демодулируются с использованием восстановленной поднесущей, не проявлялось бы заметных искажений цвета из-за фазовых артефактов. В некоторых приложениях для создания поднесущей, которая полностью бы соответствовала спецификациям стандартов NTSC, PAL или SECAM, требуется использовать системные часы, это требование, как правило, является еще более строгим, чем требования типовых телевизионных приемников. Например, в спецификации SMPTE для стандарта NTSC требуется точность поднесущей 3×10^{-6} с максимальной кратковременной нестабильностью 1 нс на одну строку и максимальным долговременным уходом частоты 0,1 Гц в секунду.

В приложениях, где восстановленные значения времени системных часов для генерации цветовой поднесущей не используются, они могут продолжать использоваться для генерации частоты следования пикселов видеосигнала, и они могут использоваться для генерации частоты дискретизации аудиосигнала. Для этих частот имеются собственные требования к стабильности, зависящие от тех предположений, которые были сделаны относительно дисплея приемника, и от допустимого сдвига звуковой частоты, или "коэффициента детонации", на выходе декодера.

В приложениях, где каждый видеокадр и каждый отсчет аудиосигнала не обязательно воспроизводится только один раз, т. е. допускается "проскальзывание" видеокадров и отсчетов аудиосигнала, к системным часам могут предъявляться относительно менее жесткие требования по точности и стабильности. Такой тип декодера может не обеспечивать точную синхронизацию воспроизведения изображения и звука, и результирующее воспроизведение аудио и видеосигналов может не иметь такого же качества, как в точно синхронизированных декодерах.

Выбор требований к точности и стабильности восстановленного времени системных часов зависит от приложения. В последующих разделах рассматривается определенное выше наиболее строгое требование, т. е. требование для случая, когда значения времени системных часов должны использоваться для синхронизации цветовой поднесущей.

D.0.6 Значения времени системных часов, используемые для генерации цветовой поднесущей

Требования к структуре декодера могут быть определены из требований к результирующей поднесущей и максимального значения нестабильности сети, которое должно быть приемлемым. Аналогично, если требования к качественным показателям системных часов и возможности декодера известны, то может быть определена максимальная допустимая нестабильность сети. Хотя, определение этих требований выходит за рамки настоящей Рекомендации | Международного стандарта, здесь приводятся цифры, необходимые для определения спецификации одного из вариантов реализации системы, что позволило уточнить описание проблемы и иллюстрирует типовой подхода к проектированию.

При использовании для восстановления частоты системных часов схемы ФАПЧ, показанной на рисунке D.2, восстановленные значения времени системных часов должны удовлетворять требованиям к наихудшему случаю отклонения частоты от номинального значения, измеренному в единицах 1×10^{-6} (одна миллионная), и наихудшему случаю скорости уходя частоты, измеренному в единицах (1×10^{-6}) /с (одна миллионная в секунду). Размах (между пиковыми значениями) нескорректированной нестабильности синхронизации имеет значение, которое может быть определено в миллисекундах. Нестабильность синхронизации сети в такой схеме ФАПЧ изображается на диаграмме в виде коэффициента ошибки **e**, и, так как ФАПЧ действует на нестабильность на своем входе как фильтр нижних частот (ФНЧ), наихудший вариант ее воздействия на выходной сигнал частотой 27 МГц наблюдается, когда на вход ФНЧ воздействует кусочно-постоянная функция синхронизации PCR с максимальной амплитудой. Тогда значение **e** имеет максимальную амплитуду, равную размаху нестабильности, что численно равно нестабильности, умноженной на 2^{33} в базовой части кода SCR или PCR. Максимальная скорость изменения выходного сигнала ФНЧ равна **f**, при наличии такого максимального значения **e** на его входе, она непосредственно определяет максимальную скорость ухода частоты выходного сигнала от номинального значения 27 МГц. ФНЧ может быть определен для любого заданного максимального значения **e** и максимальной скорости изменения **f**. Однако, с уменьшением коэффициента передачи или частоты среза ФНЧ, увеличивается время, необходимое для того, чтобы ФАПЧ синхронизировалась по частоте, описанной в полях SCR или PCR. Могут быть реализованы схемы ФАПЧ с очень большими значениями постоянных времени за счет применения цифровых ФНЧ, и, возможно, методов аналоговой фильтрации. При использовании цифровых ФНЧ, когда входным сигналом аналогового ГУН является частота **f**, она квантуется цифро-аналоговым преобразователем, размер шага которого следует учитывать при расчете максимального времени изменения частоты выходного сигнала.

Для того чтобы **e** превратилось в значение, которое стремится к нулю, коэффициент усиления разомкнутой цепи ФАПЧ должен быт очень высоким, таким, какой может быть реализован в виде функции интегратора фильтра нижних частот в цепи ФАПЧ.

При данном требовании к точности может быть целесообразным создавать ФАПЧ таким образом, чтобы исходная рабочая частота ФАПЧ удовлетворяла этому требованию к точности выходной частоты. В таком случае исходная частота 27 МГц до синхронизации в ФАПЧ, является достаточно точной для того, чтобы удовлетворять указанному требованию к выходной частоте. Если бы не тот факт, что буферы декодера, в конце концов, будут переполняться или оставаться недозаполненными, эта исходная частота системных часов была бы достаточной для долговременной работы. Однако, с того момента времени, когда декодер начинает принимать и декодировать данные, и до того момента, когда системные часы синхронизируются по времени и частоте с данными, которые представлены в принятых полях SCR или PCR, принимаемые данные поступают в буферы со скоростью, отличной от той, с которой они были выделены из потока, или, что равнозначно, декодер выделяет модули доступа в моменты времени, которые отличаются от моментов, указанных в модели декодера конечной системы (STD). Буферы декодера будут продолжать быть заполненными в большей или меньшей степени, чем это предусмотрено моделью STD, в соответствии с траекторией изменения восстановленной частоты системных часов относительно частоты системных часов кодера. В зависимости от относительной частоты сигнала ГУН и частоты системных часов кодера, степень заполненности буфера декодера будет либо увеличиваться, либо уменьшаться. Предположим, что это соотношение неизвестно, тогда для того чтобы обеспечить возможность работы в любом из указанных случаев, в декодере потребуется дополнительная буферизация данных. Декодер должен быть создан таким образом, чтобы, во избежание недозаполнения буфера, он имел возможность задерживать все операции декодирования на время, которое, как минимум, равно

времени, необходимому для дополнительной буферизации, которая предусмотрена для случая, когда исходная частота ГУН больше частоты системных часов кодера. Если исходная частота ГУН имеет точность, недостаточную для удовлетворения указанных требований к точности, то ФАПЧ должна достигать синхронизированного состояния до начала декодирования, и в этом случае появляется иной набор соображений о том, как должна работать ФАПЧ в течение этого времени, и о том, какой объем дополнительной буферизации, и какая величина статической задержки являются приемлемыми.

Кусочно-постоянная функция нестабильности входного сигнала синхронизации, которая создает кусочно-постоянную функцию коэффициента ошибки **e** в цепи ФАПЧ, показанной на рисунке D.2, должна создавать такую величину частоты выходного сигнала **f,** которая, когда она умножается на коэффициент усиления ГУН, имела бы меньшую максимальную скорость изменения, чем скорость изменения частоты, указанная в спецификации. Коэффициент усиления ГУН определяется в единицах величины изменения выходной частоты относительно изменения входного сигнала управления. Дополнительным ограничением, накладываемым на ФНЧ в ФАПЧ, является ограничение статического значения **e,** когда цепь синхронизирована для того, чтобы ограничить объем дополнительной буферизации и величину статической задержки декодирования, которые должны быть реализованы. Это значение минимально, когда ФНЧ имеет очень высокий коэффициент усиления по постоянному току.

Практически реализуемыми могут быть цепи восстановления времени системных часов, которые некоторым образом отличаются от того, что показано на рисунке D.2. Например, можно реализовать цепь управления, используя на ГУН, а генератор с числовым программным управлением (ЧПУ), где в генераторе с ЧПУ применяется генератор фиксированной частоты, и, для того чтобы синхронизировать время декодирования и воспроизведения, к обычному периодическому сигналу на его выходе добавляются (или удаляются) тактовые циклы. При таком подходе, если он используется для композитных видеосигналов, могут встречаться некоторые трудности, поскольку в нем наблюдается тенденция либо создавать заметные сдвиги фазы поднесущей, либо вносить нестабильность синхронизации строчной или кадровой разверток. Одним из возможных подходов является регулировка периода кадровой развертки в начале периода вертикального гашения, при которой фаза цветовой поднесущей остается неизменной.

Итак, в зависимости от значения, определенного требованиями, может быть или не быть разумно создать такой декодер, который восстанавливает частоту системных часов с достаточной точностью и стабильностью, поддерживая при этом желаемые размеры буферов декодеров и добавляя задержку декодирования.

D.0.7 Восстановление компонентных видео и аудиосигналов

Если на выходе декодера создается компонентный видеосигнал, то требования к точности и стабильности синхронизации, как правило, менее строгие, чем в случае композитного видеосигнала. Типичный допуск по частоте определяется допуском, который может быть приемлем в цепях развертки дисплея, и допуск по стабильности определяется необходимостью не допускать заметных перестановок изображения на дисплее.

Применяются те же принципы, что были описаны выше, однако, существуют также и конкретные требования, которые, как правило, выполнить более просто.

Скорость восстановления отсчетов аудиосигналов тоже соответствует тем же принципам; однако, требование к стабильности определяется величиной допустимых кратковременных и долговременных изменений частоты отсчетов. С применением описанного в предыдущем подразделе подхода на основе ФАПЧ, можно добиться очень малого кратковременного отклонения, а более длительное отклонение частоты обозначается как изменение воспринимаемого основного тона. Повторим еще раз, если для этого изменения установлены конкретные границы, то могут быть определены и конкретные требования к варианту реализации декодера.

D.0.8 Проскальзывание кадров

В некоторых приложениях, где точной синхронизации декодера не требуется, системные часы декодера могут не подстраивать его рабочую частоту в соответствии с частотой, представленной в принятых полях SCR (или PCR); вместо этого декодер может иметь автономный задающий генератор частоты 27 МГц. Однако при этом частота STC декодера продолжает синхронизироваться с частотой принятых данных. В таком случае значение STC должно, при необходимости, регулярно обновляться, для того чтобы соответствовать принятым значениям SCR. Обновление величины STC при приеме значений SCR приводит к нарушению непрерывности значений STC. Амплитуда этих нарушений непрерывности зависит от разницы между частотой 27 МГц декодера и частотой 27 МГц кодера, т. е. той частотой, которая представлена принятыми значениями SCR, и от интервала времени между последовательно принятыми значениями SCR или PCR. Поскольку частота 27 МГц системных часов декодера не синхронизирована с частотой принятых данных, она не может использоваться для генерации отсчетов видео или аудио сигналов, если при этом предполагается поддержание точной синхронизации воспроизведения каждого модуля представления видео или аудиосигнала, воспроизведение их только один раз и сохранение одинаковой скорости воспроизведения видео или аудиосигнала в кодере и в декодере, с точной синхронизацией аудио и видеосигналов. Существует множество возможностей реализации систем декодирования и воспроизведения, в которых используется эта структура.

При одном из вариантов реализации, видеокадры и отсчеты аудиосигнала декодируются в моменты времени, указанные значениями STC декодера, при этом они представляются в несколько различные моменты времени в соответствии с созданными в месте приема отсчетами тактовой частоты. В зависимости от соотношения отсчетов тактовой частоты декодера и отсчетов тактовой частоты кодера, видеокадры и отсчеты аудиосигнала могут в некоторых случаях воспроизводиться несколько раз, или не воспроизводиться совсем; это явление называется "проскальзыванием кадров" или "проскальзыванием отсчетов" в случае аудиопотока. Этот механизм может вносить в сигнал заметные искажения. Синхронизация видео и аудиосигналов, как правило, будет не точной из-за наличия интервалов времени, в течение которых видеокадры и, возможно модули представления аудиосигнала будут воспроизводиться повторно или, наоборот, удаляться. В зависимости от конкретного варианта реализации, в декодере, как правило, требуется дополнительная буферизация для кодированных данных или декодированных данных. Декодирование может быть выполнено непосредственно перед воспроизведением, и не обязательно в то время, которое указано значением STC декодера, либо декодированные модули представления могут быть сохранены для отложенного и, возможно, повторного воспроизведения. Если декодирование выполняется во время воспроизведения, необходим некоторый механизм исключения процесса воспроизведения некоторых видеокадров и отсчетов аудиосигнала, без создания проблем процессу декодировании последующих данных, которые были закодированы с предсказанием.

D.0.9 Сглаживание нестабильности сети

В некоторых приложениях между сетью и декодером можно ввести специальный механизм, обеспечивающий уменьшение степени нестабильности, которая вносится сетью. Реализуем ли такой подход, зависит от типов принимаемых потоков, а также от величины и типа ожидаемой нестабильности.

И Транспортный поток, и Программный поток в своем синтаксисе указывают скорость, с которой, как ожидается, поток будет поступать на вход декодера. Эти указанные скорости не являются точными и не могут использоваться для точного восстановления синхронизации потока данных. Однако, они могут быть полезными в виде части механизма сглаживания.

Например, Транспортный поток может быть принят из сети таким образом, что данные доставляются в импульсном режиме. Принятые данные можно сохранить в буфере, и передавать их из буфера в декодер с примерно постоянной скоростью так, чтобы буфер всегда оставался наполовину заполненным.

Однако, поток, передаваемый с переменной скоростью, не может доставляться с постоянной скоростью, и для потоков с переменной скоростью буфер сглаживания не всегда будет наполовину заполненным. Постоянная средняя задержка сигнала в буфере требует, чтобы степень заполненности буфера изменялась с изменением скорости передачи. Скорость, с которой данные должны извлекаться из буфера, и подаваться на декодер, можно приблизительно оценить с использованием информации о скорости передачи потока данных. В Транспортных потоках предполагаемая скорость определяется значениями в полях PCR и количеством байтов Транспортного потока между ними. В Программных потоках предполагаемая скорость явно указана в виде значения Program_mux_rate, хотя, как определено в настоящей Рекомендации | Международном стандарте, эта скорость может упасть до нуля в точках размещения SCR, т. е. если поле SCR принимается до того времени, когда ожидался бы его прием, если данные доставляются с указанной скоростью.

В случае потоков данных с переменной скоростью правильная степень заполнения буфера сглаживания изменяется во времени, и не всегда может быть точно определена из информации о скорости. При альтернативном подходе для измерения времени, в которое данные поступают в буфер, и для регулировки времени, когда данные покидают буфер, могут использоваться значения SCR или PCR. Может быть разработана цепь управления, которая обеспечивает постоянную среднюю задержку в буфере. Можно видеть, что такой вариант реализации аналогичен цепи управления, показанной на рисунке D.2. Качественные показатели, достигаемые в результате введения такого механизма сглаживания перед декодером, могут быть получены также при последовательном включения нескольких цепей ФАПЧ для восстановления синхронизации. За счет объединения влияния фильтров нижних частот и последовательно включенных цепей ФАПЧ будет повышена стабильность синхронизации принимаемого сигнала.

Приложение E

Приложения передачи данных

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

E.0 Общие соображения

- Для передачи данных, а также видео и аудио сигналов будет использоваться Транспортный мультиплексированный поток, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.
- Элементарные потоки данных не являются непрерывными, каковыми являются аудио и видеопотоки в радиовещательных приложениях.
- Несмотря на то, что уже вполне возможно идентифицировать начало пакета PES, не всегда можно идентифицировать окончание пакета PES как начало следующего пакета PES, поскольку возможна потеря одного или нескольких Транспортных пакетов, в которых передаются пакеты PES.

E.1 Предложение

Подходящим решением является передача следующего пакета PES сразу же после связанного с ним пакета PES. Когда более нет пакетов PES, предназначенных для передачи, может быть передан пакет PES без полезной нагрузки.

В таблице E.1 показан пример такого пакета PES.

Таблица E.1 – Пример заголовка пакета PES

Поля заголовка пакета PES	Значения
packet_start_code_prefix	0x000001
stream_id	назначенное
PES_packet_length	0x0003
'10'	'10'
PES_scrambling_control	'00'
PES_priority	'0'
data_alignment_indicator	'0'
copyright	'0'
original_or_copy	'0'
PTS_DTS_flags	'00'
ESCR_flag	'0'
ES_rate_flag	'0'
DSM_trick_mode_flag	'0'
additional_copy_info_flag	'0'
PES_CRC_flag	'0'
PES_extension_flag	'0'
PES_header_data_length	0x00

Приложение F

Графические обозначения синтаксиса для настоящей Рекомендации | Международного стандарта

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

F.0 Введение

Данное Приложение является информативным Приложением, в котором графически представлен синтаксис Транспортного потока и Программного потока. Данное Приложение ни при каких условиях не заменяет какие-либо нормативные разделы.

Для того чтобы получить понятные рисунки, в них полностью описаны или представлены не все поля. Поля, которые зарезервированы, могут быть пропущены или указаны областями без дополнительных подробностей. Продолжительность полей указана в битах.

F.0.1 Синтаксис Транспортного потока

См. рисунок F.1.

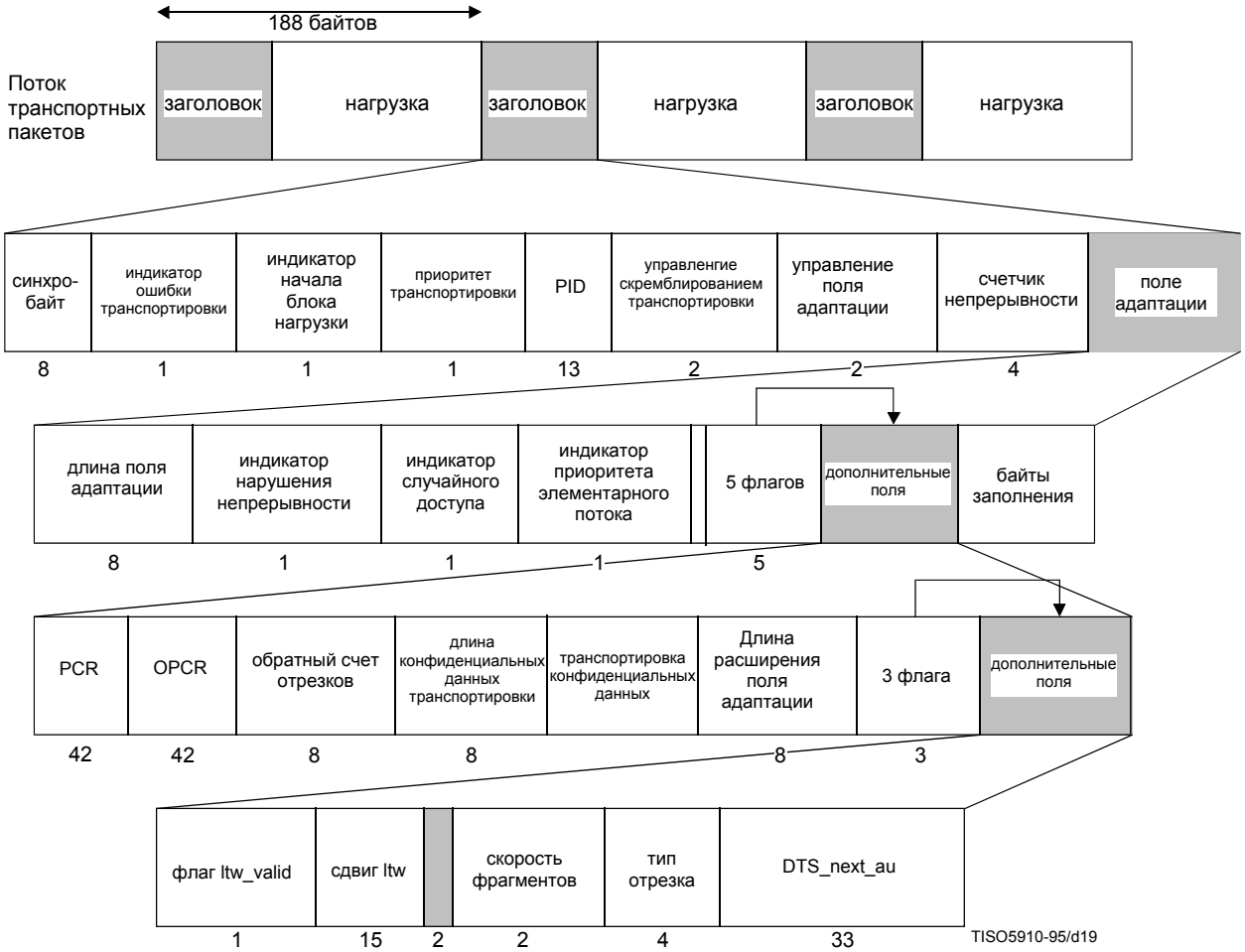


Рисунок F.1 – Диаграмма синтаксиса Транспортного потока

F.0.2 Пакет PES

См. рисунок F.2.

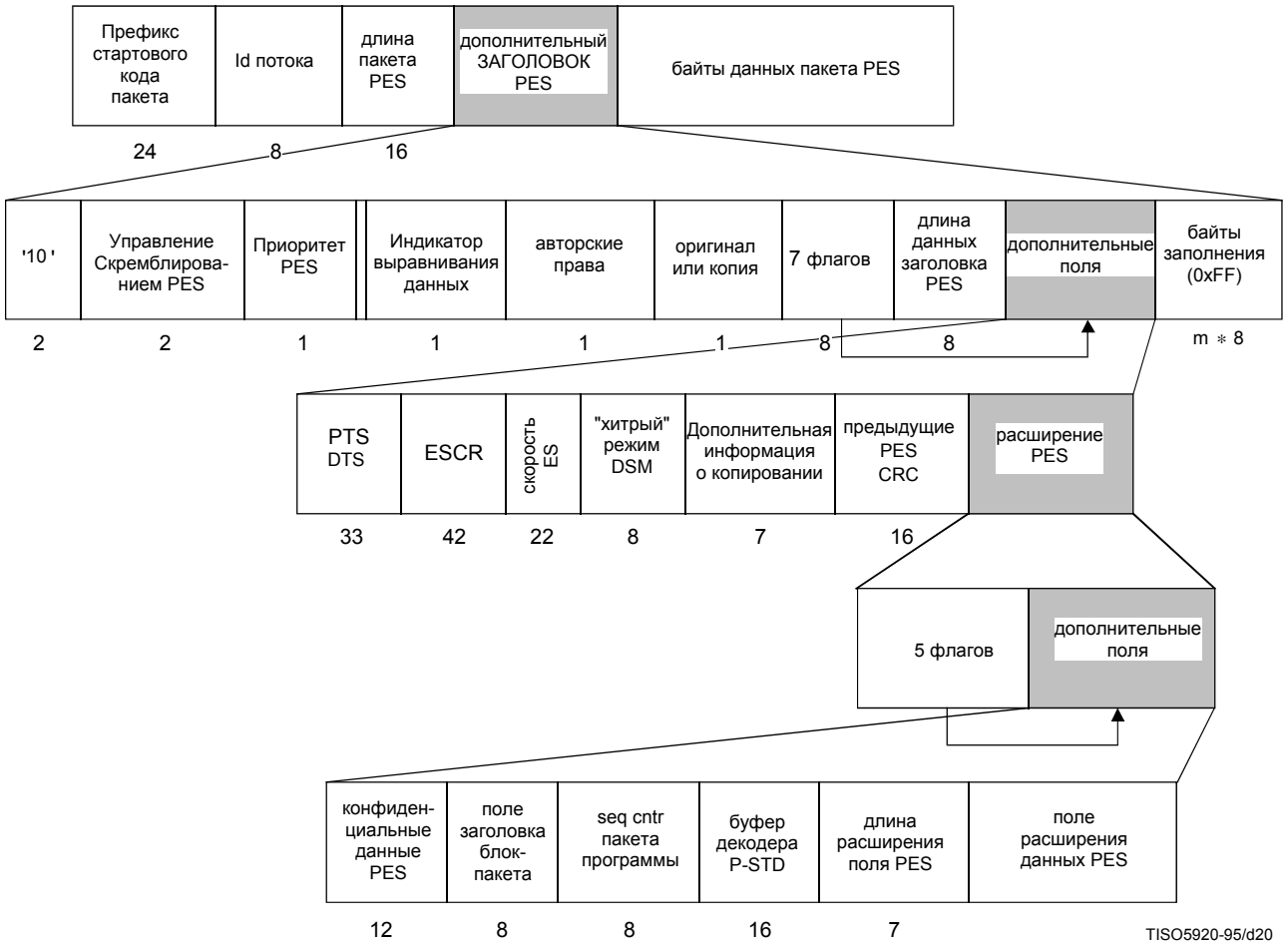


Рисунок F.2 – Диаграмма синтаксиса пакета PES

F.0.3 Секция ассоциации программы

См. рисунок F.3.

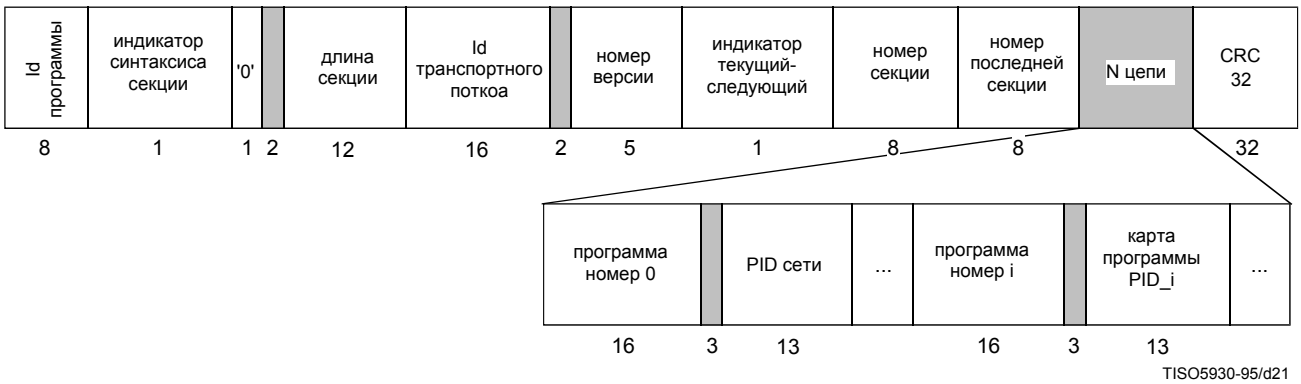


Рисунок F.3 – Диаграмма секции ассоциации программы

F.0.4 Секция условного доступа (CA)

См. рисунок F.4.

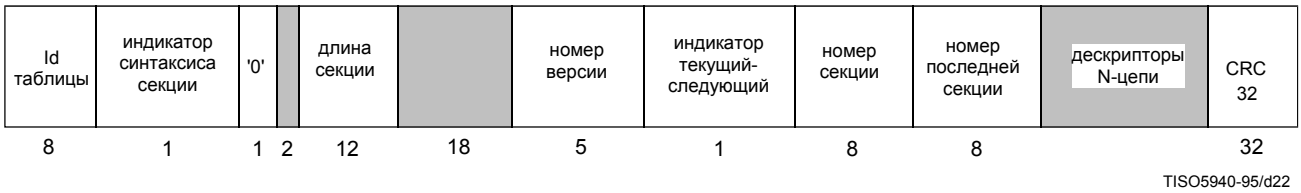


Рисунок F.4 – Диаграмма секции условного доступа

F.0.5 Секция карты программы транспортного потока (TS)

См. рисунок F.5.

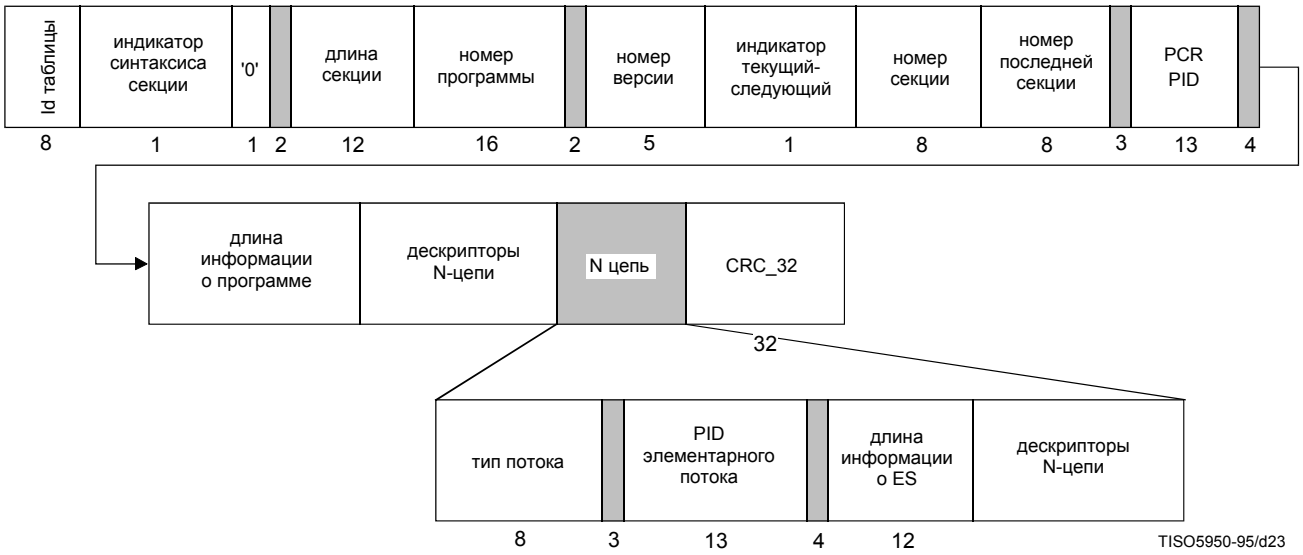


Рисунок F.5 – Диаграмма секции карты программы TS

F.0.6 Конфиденциальная секция

См. рисунок F.6.

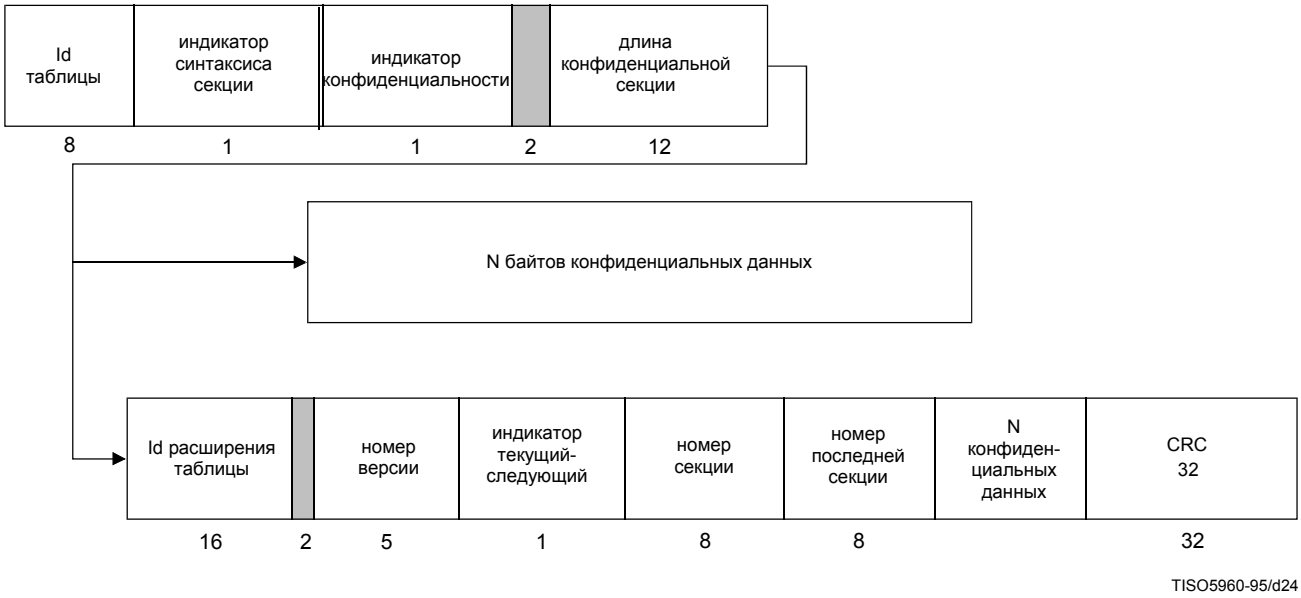


Рисунок F.6 – Диаграмма конфиденциальной секции

F.0.7 Программный поток

См. рисунок F.7.

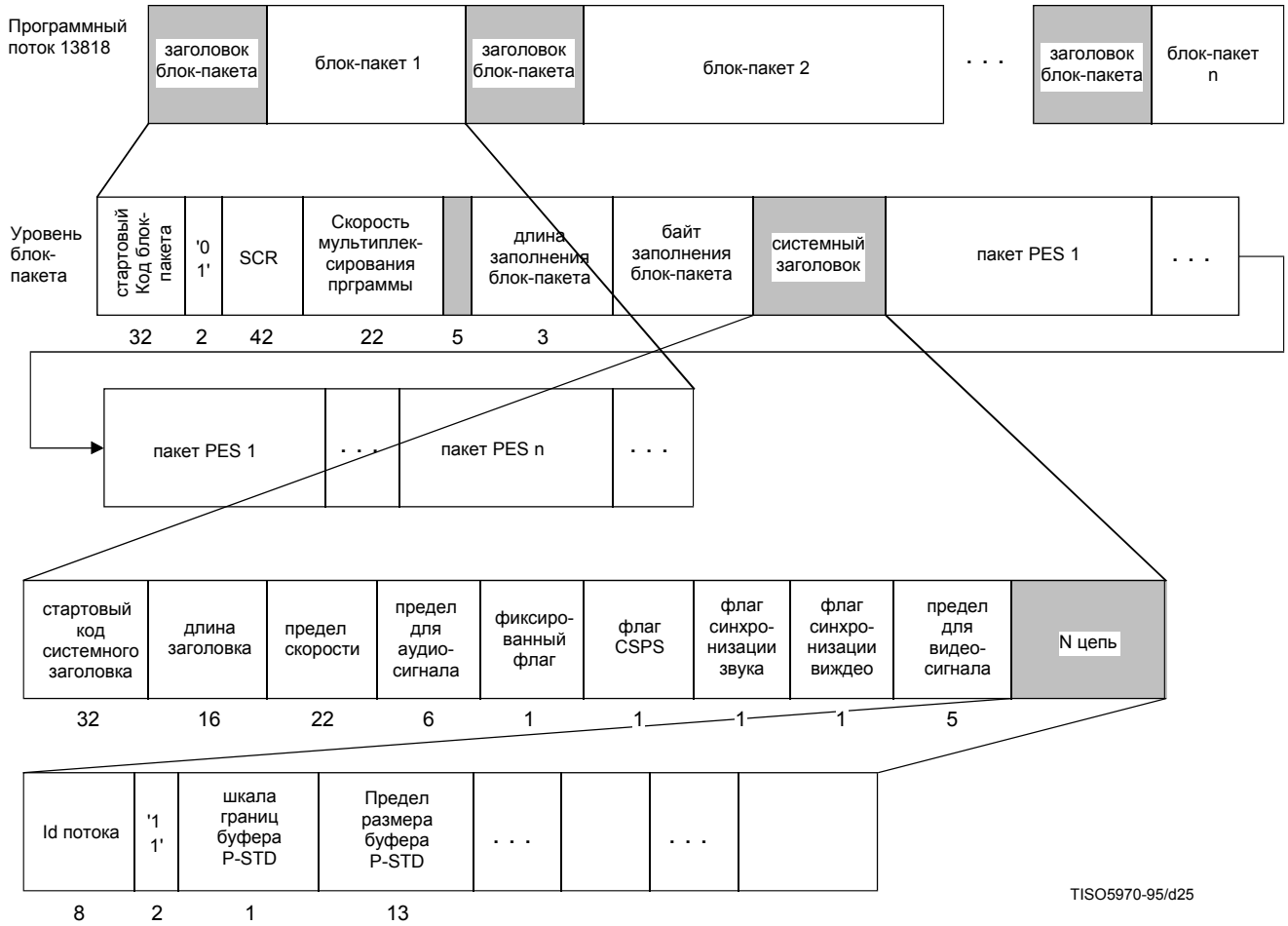


Рисунок F.7 – Диаграмма Программного потока

F.0.8 Карта Программного потока

См. рисунок F.8.

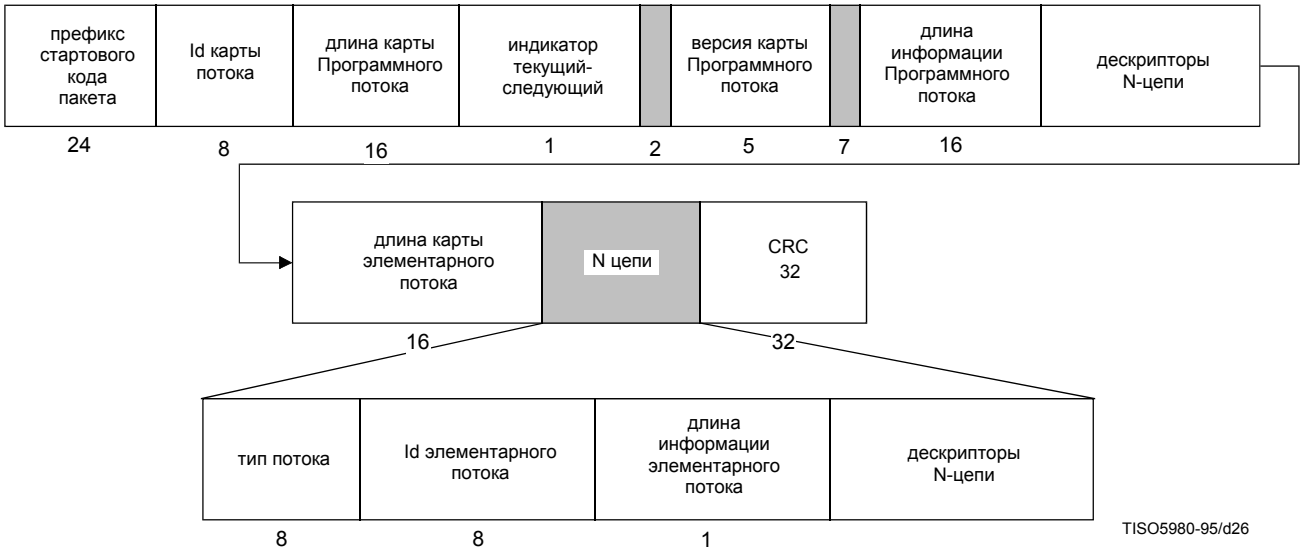


Рисунок F.8 – Диаграмма карты Программного потока

Приложение G

Общая информация

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

G.0 Общая информация

G.0.1 Эмуляция синхробайта

При выборе значений PID рекомендуется, избегать периодической эмуляции синхробайтов. Такая эмуляция может возникать в поле PID или как комбинация поля PID и флагов соседних с ним полей. Рекомендуется разрешить эмуляцию синхробайта так, чтобы он появлялся в одной и той же позиции заголовка пакета в течение не более 4 последовательных транспортных пакетов.

G.0.2 Состояние пропущенного кадра и процесс декодирования

Предположим, что воспроизводимая последовательность содержит только I- и P- кадры. Обозначим следующий кадр, который должен быть декодирован как picture_next, а кадр, декодируемый в настоящий момент, как picture_current. Из-за того, что видеокодер может пропускать видеокадры, может случиться так, что не все биты кадра picture_next находятся в буферах декодера STD EB_n или B_n, когда приходит время извлекать эти биты для декодирования и воспроизведения. При возникновении такого случая, ни один бит из буфера не удаляется, и повторно воспроизводится кадр picture_current. Когда приходит время воспроизведения следующего кадра, если теперь в буфере EB_n или B_n находятся оставшиеся биты кадра picture_next, то из буфера удаляются все биты кадра picture_next и воспроизводится кадр picture_next. Если в буфере EB_n или B_n находятся не все биты кадра picture_next, то повторяется вышеописанный процесс повторного воспроизведения кадра picture_current. Этот процесс повторяется до тех пор, пока не сможет быть воспроизведен кадр picture_next. Отметим, что если метка времени PTS предшествует кадру picture_next в двоичном потоке, она будет неверна на несколько целых интервалов воспроизведения кадра, что само по себе может зависеть от некоторых параметров и должно быть проигнорировано.

Когда бы не возникла описанная выше ситуация с пропущенным кадром, требуется, чтобы кодер вводил метку PTS до того кадра, который должен быть декодирован после кадра picture_next. Это позволит декодеру сразу же проверить, правильно ли он воспроизводит принятую последовательность кадров.

G.0.3 Выбор значений PID

Желательно, чтобы в приложениях использовались небольшие численные значения PID (избегая зарезервированных значений, определенных в таблице 2-4) и чтобы значения максимально возможно группировались вместе.

G.0.4 Эмуляция стартового кода PES (start_code)

Три последовательных байта, имеющих префикс стартового кода пакета packet_start_code_prefix = 0x000001, которые при объединении с четвертым байтом, могут эмулировать четыре байта заголовка PES_packet_header в том месте потока, где этот заголовок появляться не должен.

Таким образом, в элементарных видеопотоках так называемая эмуляция стартового кода не возможна. Она возможна в элементарных аудиопотоках и элементарных потоках данных. Она также возможна на границе полей PES_packet_header и PES_packet_payload, даже если PES_packet_payload представляет собой видеосигнал.

Приложение H

Конфиденциальная информация

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

H.0 Конфиденциальная информация

Конфиденциальная информация представляет собой любые данные пользователя, которые не кодируются в соответствии со стандартом, определенным МСЭ-Т | ИСО/МЭК, и описанным в настоящей Спецификации. Содержание этих данных в настоящей Рекомендации | Международном стандарте не определяется и не должно определяться в будущем. Декодер STD, определенный в настоящей Спецификации, после процесса

демультиплексирования не производит над конфиденциальной информацией действий. Каждая сторона может самостоятельно определять STD для потоков конфиденциальной информации.

Конфиденциальная информация может передаваться в следующих областях синтаксиса, описанного в Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

1) *пакет Транспортного потока таблица 2-2*

   Байты данных синтаксиса transport_packet() могут содержать конфиденциальную информацию. Конфиденциальная информация, передаваемая в таком формате, называется конфиденциальной информацией пользователя в потоке типа stream_type (таблица 2-34). Допускается, чтобы содержащаяся в пакетах Транспортного потока конфиденциальная информация включала в себя поля adaptation_field().

2) *Поле адаптации Транспортного потока таблица 2-6*

   О наличии любых дополнительных байтов конфиденциальной информации private_data_bytes в поле adaptation_field() сообщает флаг transport_private_data_flag. Число байтов конфиденциальной информации private_data_bytes, по определению, ограничивается семантикой поля adaptation_field_length, в котором длина поля adaptation_field_length не должна превышать 183 байтов.

3) *Пакет PES таблица 2-21*

   Существует две возможности передачи конфиденциальной информации в пакетах PES. Первая возможность – в заголовке PES_packet_header, в пределах дополнительных 16 байтов PES_private_data. О наличии этого поля сообщает флаг PES_private_data_flag. О наличии флага PES_private_data_flag сообщает флаг PES_extension_flag. Если эти байты присутствуют, то они вместе с соседними полями не должны эмулировать префикс packet_start_code_prefix.

   Вторая возможность – в поле PES_packet_data_byte. ЕЕ можно назвать передачей конфиденциальной информации в пакетах PES потока типа stream_type (таблица 2-34). Эта категория конфиденциальной информации может быть разделена на две: поток private_stream_1 обозначает конфиденциальную информацию в пакетах PES, которые соответствуют синтаксису PES_packet() так, что присутствуют все поля до поля PES_header_data_length, включительно, но не ограничиваясь ими. Поток private_stream_2 обозначает конфиденциальную информацию в пакетах PES, в которых должны присутствовать только первые три поля, следующие после байтов PES_packet_data_bytes, содержащих конфиденциальную информацию.

   Отметим, что пакеты PES существуют как в Программных, так и в Транспортных потоках, следовательно, как в Программных, так и в Транспортных потоках существуют потоки private_stream_1 и private_stream_2.

4) *Дескрипторы*

   Дескрипторы существуют и в Программных, и в Транспортных потоках. Диапазон конфиденциальных дескрипторов может быть определен пользователем. Эти дескрипторы должны начинаться с полей descriptor_tag и descriptor_length. Для конфиденциальных дескрипторов значение поля descriptor_tag может принимать значения от 64 до 255, как определено в таблице 2-45. Эти дескрипторы могут быть размещены в карте program_stream_map() (таблица 2-34), секции CA_section() таблицы 2-32, секции TS_program_map_section() (таблица 2-33) и в любой секции private_section() (таблица 2-35).

   В частности, байты private_data_bytes появляются также в дескрипторе CA_descriptor().

5) *Конфиденциальная секция*

   Конфиденциальная секция private_section (таблица 2-35) предоставляет дополнительные возможности для передачи конфиденциальной информации также в двух формах. Этот тип элементарного потока может быть идентифицирован полем stream_type (таблица 2-34) как секции private_data в PSI. Один из типов секции private_section() включает в себя только пять первых определенных полей, за которыми следует конфиденциальная информация. Для этой структуры индикатор section_syntax_indicator должен быть установлен, равным '0'. Для другого типа индикатор section_syntax_indicator должен быть установлен, равным '1', и должен присутствовать полный синтаксис до last_section_number, включительно, после которого следуют байты private_data_bytes, и, в конце, CRC_32.

Приложение I

Соответствие систем и интерфейс реального времени

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

I.0 Соответствие систем и интерфейс реального времени

Соответствие Программных и Транспортных потоков Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1 определяется в понятиях нормативных спецификаций, описанных в настоящей Рекомендации | Международном стандарте. Помимо других требований, эти спецификации включают в себя декодер конечной системы (T-STD и P-STD), который определяет работу идеального декодера, когда поток является входным сигналом такого декодера. Эта модель и связанные с ней проверки не содержат информации, касающейся качества доставки потока в реальном времени, за исключением точности частоты системных часов, которая определяется Транспортным потоком и Программным потоком. Все Транспортные потоки и Программные потоки должны соответствовать настоящей Рекомендации | Международному стандарту.

Кроме того, существует спецификация интерфейса реального времени для подачи на вход декодера Транспортных и Программных потоков. Настоящая Рекомендация | Международный стандарт дает возможность стандартизировать интерфейс между декодерами MPEG и адаптерами сетевых каналов или средой хранения. Влияние синхронизации каналов и неспособность практических адаптеров полностью устранить это влияние приводит к появлению отклонений от идеального времени доставки байтов. Хотя вовсе необязательно, что это интерфейс реализован во всех декодерах MPEG, те варианты реализации, который включают в себя этого интерфейс, должны соответствовать спецификациям. В настоящей Рекомендации | Международном стандарте рассматривается такая доставка в реальном времени Транспортных потоков и Программных потоков на декодеры, при которой гарантируется, что буферы декодеров для кодированных данных не будут ни переполняться, ни оставаться недозаполненными, а также гарантируется, что декодеры будут способны обеспечить восстановление частоты системных часов с качеством, требуемым для данных приложений.

Интерфейс MPEG реального времени определяет максимально допустимый размер отклонения от идеального времени доставки байтов, которое указано в полях Эталонного времени программы (PCR) и Эталонного времени системы (SCR), закодированных в данном потоке.

Приложение J

Взаимодействие систем, вносящих нестабильность синхронизации, с декодерами MPEG

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

J.0 Введение

В настоящем Приложении выражение "системный поток" будет использоваться для обозначения как Транспортных потоков, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, так и Программных потоков, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Когда используется термин "STD", он обозначает P-STD (программный декодер конечной системы) для Программных потоков и T-STD (транспортный декодер конечной системы) для Транспортных потоков.

Предполагаемое время доставки байтов системного потока может быть рассчитано в результате анализа потока. Системный поток является соответствующим, если он может быть декодирован при помощи STD, который представляет собой математическую модель идеального декодера. Если соответствующий системный поток передается по сети, вносящей нестабильность, то реальное время доставки байта может значительно отличаться от предполагаемого времени доставки байтов. В таких случаях идеальный декодер может оказаться неспособным декодировать системный, поскольку нестабильность может привести к переполнению или недозаполнению буферов, и восстановить временную базу может оказаться очень трудным. Хорошим примером такой вносящей нестабильность сети является ATM.

Целью настоящего Приложения является обеспечить понимание работы блоков, связанных с передачей системных потоков по сетям, вносящим нестабильность. Вероятно, для нескольких типов сетей, включая ATM, будут разработаны модели соответствия, зависящие от конкретных сетей. В определении таких моделей общую роль могут играть декодер STD и определение интерфейса реального времени. База для разработки моделей соответствия систем описана в разделе J.1.

В разделе J.2 рассмотрено три примера кодирования сети, позволяющего создавать сетевые адаптеры, сглаживающие нестабильность. В первом примере предполагается системный поток с постоянной скоростью передачи, и для сглаживания нестабильности используется обработка в порядке поступления (FIFO). Во втором примере уровень адаптации сети включает в себя метки времени, позволяющие сгладить нестабильность. В последнем примере предполагается, что на всей длине передачи доступен общий генератор тактовой частоты, и он используется для обеспечения сглаживания нестабильности.

В разделе J.3 приведены два примера вариантов реализации декодера, в которых может быть учтена нестабильность, вносимая сетью. В первом примере между выходом сети и декодером MPEG-2 введен сетевой адаптер, сглаживающий нестабильность. Предполагается, что декодер MPEG-2 соответствует спецификации интерфейса MPEG-2 реального времени. Для этого интерфейса требуется декодер MPEG-2 с большим допуском на нестабильность, чем в идеальном декодере STD. Этот сетевой адаптер обрабатывает входящий нестабильный двоичный поток и получает на выходе системный поток, у которого реальное время доставки байтов соответствует спецификации реального времени. Первый пример обсуждается в разделе J.3.1. Для некоторых приложений подход на основе сетевого адаптера может быть чересчур дорогостоящим, поскольку для него требуется два этапа обработки. Следовательно, во втором примере функции устранения нестабильности и декодирования MPEG-2 объединяются. Промежуточная обработка на устройстве устранения нестабильности пропускается, таким образом, требуется только один этап восстановления частоты системных часов. Декодеры, которые выполняют интегральное устранение нестабильности и декодирование, в настоящем Приложении называются интегральными сетевыми декодерами, или просто интегральными декодерами. Интегральные декодеры обсуждаются в разделе J.3.2.

Для того чтобы создать либо сетевые адаптеры, либо интегральные декодеры, необходимо сделать предположение относительно максимальной величины размаха нестабильности в сети. Для обеспечения взаимодействия должно быть определено предельное значение размаха нестабильности для каждого соответствующего типа сети.

J.1 Модели соответствия систем

Один из способов моделирования передачи системного потока по сети, вносящей нестабильность, показан на рисунке J.1.

Системный поток является входным сигналом для устройства кодирования, предназначенного для данной сети, которое преобразует системный поток в формат, определяемый сетью. Частью этого формата может быть информация, которая помогает устранить нестабильность на выходе сети. Сетевой декодер состоит из декодера, определяемого сетью, и декодера, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Предполагается, что декодер, соответствующий Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, отвечает спецификации реального времени, и может иметь точно такую же архитектуру, что и STD, причем предусмотрены большие размеры соответствующих буферов для обеспечения большего допуска на нестабильность. Декодер, определяемый сетью, удаляет данные, не соответствующие Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, добавленные кодером, определяемым сетью, и устраняет нестабильность на выходе сети. Выходным сигналом декодера, определяемого сетью, является системный поток, который отвечает спецификации реального времени

На базе вышеописанной архитектуры можно определить Оконечный сетевой декодер (NTD). Двоичным потоком, соответствующим сети, будет такой поток, который может быть декодирован при помощи NTD. Сетевой декодер будет совместимым с сетью при условии, что он может декодировать любой двоичный поток сети, который может быть декодирован в NTD. Реальный сетевой декодер может иметь или не иметь архитектуру декодера NTD.

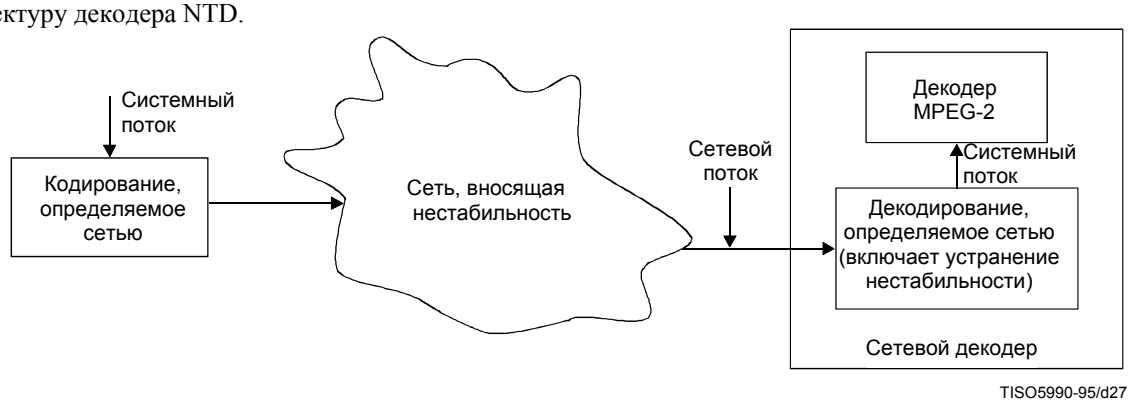


Рисунок J.1 – Передача системных потоков по сети, вносящей нестабильность

J.2 Спецификация сети для уменьшения нестабильности синхронизации

В случае системных потоков с постоянной скоростью передачи, сглаживание нестабильности может быть выполнено при помощи обработки в порядке поступления (FIFO). При этом на уровне адаптации сети не требуется дополнительной информации, которая обеспечивает поддержку устранения нестабильности. После того, как удалены байты, добавленные в процессе сетевого кодирования, данные системного потока помещаются в буфер FIFO. ФАПЧ поддерживает буфер примерно наполовину заполненным, регулируя скорость выходного сигнала в ответ на изменения в степени заполнения буфера. В этом примере достигаемая степень сглаживания нестабильности будет зависеть от размеров буфера FIFO и характеристик ФАПЧ.

На рисунке J.2 показан второй способ осуществления сглаживания нестабильности. В этом примере предполагается, что имеется поддержка, обеспечиваемая метками времени на уровне адаптации сети. Используя этот метод можно устранять нестабильность в системных потоках, как с постоянной, так и с переменной скоростью.

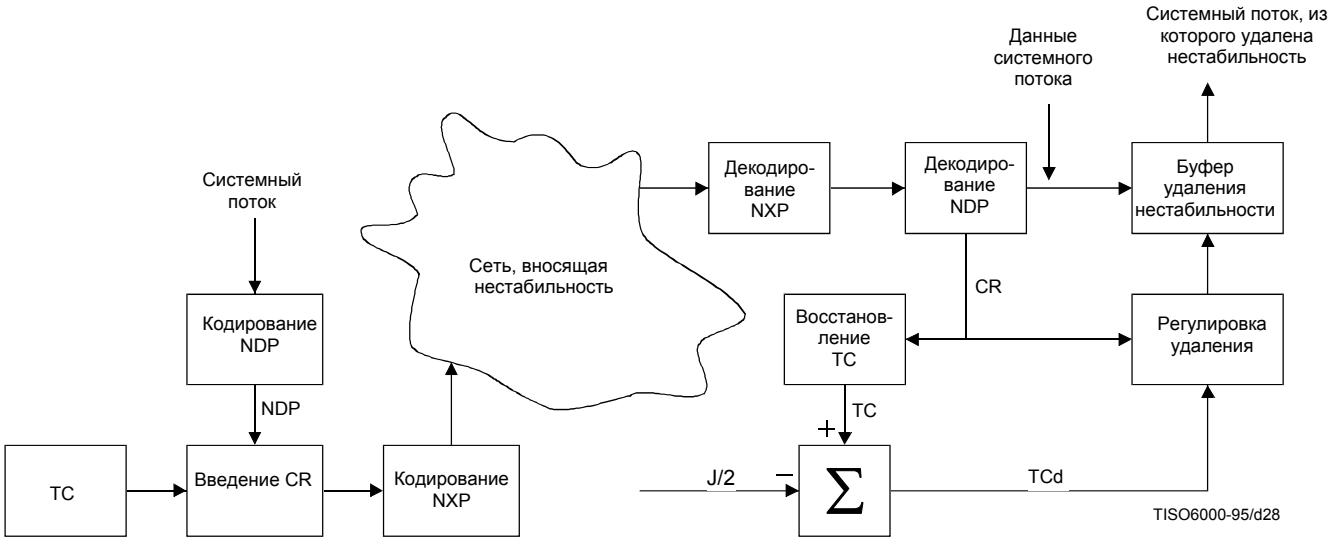


Рисунок J.2 – Устранение нестабильности с использованием меток времени сетевого уровня

Предположим, что сетевой адаптер разработан так, чтобы компенсировать размах нестабильности, равный J секунд. Предполагаемое время доставки байтов восстанавливается с использованием отсчетов Эталонного времени (CR), которые берутся из системных часов (TC). CR и TC аналогичны значениям PCR и STC. Кодер пакета данных сети (NDP) преобразует каждый пакет системного потока в пакет данных сети (NDP). Пакеты данных сети содержат поле для передачи значений CR, и текущее значение TC вводится в это поле сразу после того, как пакет NDP выходит из кодера NDP. Функция формирования транспортных пакетов сети (NXP) инкапсулирует пакеты NDP в транспортные пакеты сети. После передачи по сети декодер NDP ожидает получения значений CR, когда пакеты NDP поступают в декодер NDP. Значения CR используются для восстановления TC, например, при помощи ФАПЧ. Первый пакет MPEG-2 удаляется из буфера устранения нестабильности, когда задержанное значение TC (TCd) становится равным значению CR первого пакета MPEG-2. Последующие пакеты MPEG-2 удаляются, когда их значения CR становятся равными значению TCd.

Если не учитывать подробности варианта реализации, такие как скорость работы цепи восстановления частоты TC и спектральная чистота TC, то размер буфера устранения нестабильности зависит только от максимального размаха нестабильности, которая должна быть сглажена, и наибольшей скорости транспортировки, которая наблюдается в системном потоке. Размер буфера устранения нестабильности B_{dj} определяется как

$$B_{dj} = JR_{max},$$

где R_{max} – максимальная скорость системного потока, измеренная в битах в секунду. Когда пакеты, передаваемые по сети, испытывают номинальную задержку, буфер заполнен наполовину. Когда они испытывают задержку, равную J/2 секундам, буфер пуст, а когда они испытывают задержку, равную – J/2 секундам, буфер полон.

В качестве финального примера, в некоторых случаях на всей длине передачи доступен общий генератор тактовой частоты, и он используется для синхронизации частоты системных часов с общим генератором тактовой частоты. Сетевой адаптер может сглаживать нестабильность, используя буфер FIFO. Для восстановления исходного времени доставки байтов адаптер использует значения PCR или SCR.

J.3 Примеры реализации декодера

J.3.1 Сетевой адаптер, после которого установлен декодер MPEG-2

В таком варианте реализации сетевой адаптер, отвечающий спецификации соответствия сети, соединяется с декодером MPEG-2, соответствующим спецификации интерфейса реального времени.

J.3.2 Интегральный декодер

В примере, описанном в J.3.1, требуется два этапа обработки. Первый этап необходим для устранения нестабильности в выходном сигнале сети. Второй этап, на котором восстанавливается STC в результате обработки значений PCR или SCR, необходим для декодирования STD. Пример, показанный в настоящем подразделе, представляет собой декодер, который объединяет в единую систему функции устранения нестабильности и декодирования. Частота STC восстанавливается непосредственно с использованием искаженные нестабильностью значения PCR или SCR. Для изображения этого примера, предполагается, что передается Транспортный поток MPEG-2.

На рисунке J.3 показана работа интегрального декодера. Предполагается, что поток пакетов сети, поступающий на вход декодера, такой же, как показано на рисунке J.2.

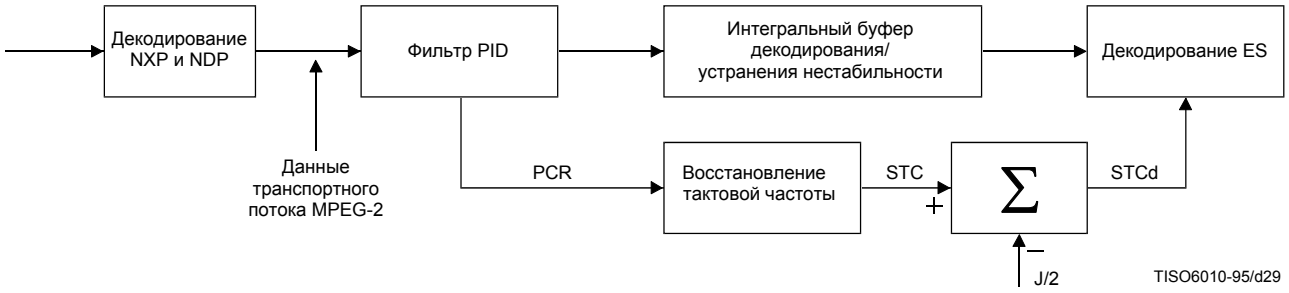


Рисунок J.3 – Интегральное устранение нестабильности и декодирование MPEG-2

Входящие пакеты сети реассемблируются в данные Транспортного потока MPEG-2 посредством функций декодирования NXP и NDP. Подверженные воздействию нестабильности пакеты Транспортного потока, соответствующего Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, затем фильтруются с целью извлечения пакетов с требуемыми значениями PID. Для изображенного случая декодированные значения PID содержат также значения PCR. Значения PCR передаются на ФАПЧ для восстановления STC. Полные пакеты с выбранными значениями PID помещаются в интегральный буфер. Положительное значение J/2 секунд вычитается из STC, в результате получается задержанное значение STC, т. е. STCd. Повторим, J – это размах нестабильности, с которой может работать декодер, разработанный для этой сети. Задержка вводится для гарантии того, что в тот момент, когда значение PTS/DTS модуля доступа становится равным текущему значению STCd, все данные, необходимые для модуля доступа, уже поступили в буфер.

Не учитывая подробности варианта реализации, такие как скорость работы цепи восстановления частоты STC и спектральная чистота STC, размер буфера равен:

$$\begin{aligned} B_{size} &= B_{dec} + B_{mux} + B_{OH} + 512 + B_j \\ &= B_n + 512 + B_j, \end{aligned}$$

где $B_j = R_{max}\, J$ и R_{max} максимальная скорость, с которой данные поступают на вход фильтра PID. В зависимости от варианта реализации, интегральная память может быть разделена на две части, как в транспортном декодере STD.

Приложение K

Объединение транспортных потоков

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

K.0 Введение

В тексте настоящего Приложения термин 'сращивание' обозначает выполняемое на транспортном уровне объединение двух различных элементарных потоков, при котором результирующий Транспортный поток полностью соответствует настоящей Рекомендации | Международному стандарту. Два этих элементарных потока могут быть созданы в различных местах и/или в различное время, и в момент своего создания могли не предназначаться для последующего объединения. В последующих разделах "старым" потоком мы будем называть непрерывный элементарный поток (видео или аудио), который в некоторой точке был заменен другим потоком ("новым"). Эта точка называется точкой объединения (стыком). Это граница между данными, принадлежащими "старому" потоку, и данными, принадлежащими "новому" потоку.

Объединение может быть бесшовным или не бесшовным:

- Точка бесшовного объединения – это точка объединения, сращивание в которой выполняется без нарушения непрерывности декодирования (см. § 2.7.6). Это значит, что время декодирования первого модуля доступа "нового" потока соответствует времени декодирования модуля доступа "старого", т. е. равно времени, которое имел бы следующий модуль доступа, если бы продолжался "старый" поток. В последующих разделах мы будем называть это время декодирования 'временем бесшовного декодирования'.
- Точка не бесшовного объединения – это точка объединения, в которой происходит нарушение непрерывности декодирования, т. е. время декодирования первого модуля доступа "нового" потока больше времени бесшовного декодирования.

  ПРИМЕЧАНИЕ. – Время декодирования, меньшее, чем время бесшовного декодирования, является запрещенным.

Сращивание разрешено на границе любого пакета Транспортного потока, пока результирующий поток остается законным. Но, в общем случае, если ничего не известно о точке начала пакета PES и точке начала модуля доступа, вводится ограничение, согласно которому разбивается не только уровень Транспортировки, но также и уровень PES, и уровень элементарного потока, и, в некоторых случаях, может потребоваться некоторая обработка для пакетов полезной нагрузки Транспортного потока. Если такой сложной операции желательно избежать, то сращивание должно быть выполнено в тех местах, где Транспортный поток имеет благоприятные свойства, эти свойства указываются наличием точек объединения.

Наличие точки объединения указывается полями splice_flag и splice_countdown (семантика этих полей описана в § 2.4.3.4). В последующих разделах пакет Транспортного потока, в котором значение поля splice_countdown достигает нуля, будет называться 'пакетом сращивания'. Точка объединения располагается сразу же после последнего байта пакета сращивания.

K.1 Различные типы точек объединения

Точка объединения может быть либо обычной точкой объединения, либо точкой бесшовного объединения.

K.1.1 Обычные точки объединения

Если поле seamless_splice_flag отсутствует, или его значение равно нулю, точка объединения является обычной. Наличие обычной точки объединения говорит только о свойствах выравнивания элементарного потока: пакет сращивания завершается в последнем байте модуля доступа, и полезная нагрузка следующего пакета Транспортного потока с одним и тем же PID начнется в заголовке пакета PES, полезная нагрузка которого начнется в точке доступа элементарного потока (или, в случае видеопотока, в коде sequence_end_code(), который следует непосредственно после точки доступа элементарного потока). Эти свойства дают возможность довольно просто выполнять операции "вырезать и вставить" на уровне транспортировки, не нарушая при этом синтаксических ограничений и гарантируя соответствие двоичного потока. Однако, она не предоставляет никакой информации относительно свойств буферизации или синхронизации. Вследствие этого, в таких точках объединения, бесшовное объединение может быть выполнено только при помощи специальных функций, или посредством анализа полезной нагрузки пакетов Транспортного потока и отслеживания состояния буфера и значений меток времени.

K.1.2 Точки бесшовного объединения

Если поле seamless_splice_flag присутствует и его значение равно единице, то информация, предоставляемая точкой объединения, указывает некоторые свойства "старого" потока. Эта информация не направляется в декодеры. Ее первоочередная цель – обеспечить бесшовное объединение. Такая точка объединения называется точкой бесшовного объединения. В ней доступная следующая информация:

- Время бесшовного объединения, которое кодируется в виде значения DTS в поле DTS_next_AU. Это значение DTS выражено в той временной базе, которая действует в пакете сращивания.
- Для элементарного видеопотока – ограничения, которые применялись к "старому" потоку в момент его создания, предназначенные для обеспечения бесшовного объединения. Эти условия указываются значением поля splice_type, в таблице, соответствующей профилю и уровню видеопотока.

Отметим, что точка бесшовного объединения может использоваться как обычная точка объединения, если отбросить эту дополнительную информацию. Эта информация может также использоваться, если она считается полезной для выполнения не бесшовного объединения, или для целей, отличных от сращивания.

K.2 Характеристики декодера при объединениях

K.2.1 При не бесшовных объединениях

Как описано выше, не бесшовное объединение – это объединение, которое приводит к нарушению непрерывности декодирования.

Следует отметить, что при таком объединении должны соблюдаться ограничения, относящиеся к нарушению непрерывности декодирования (см. § 2.7.6). В частности:

- метка времени PTS должна быть закодирована для первого модуля доступа "нового" протока (за исключением случая работы в "хитром" режиме, или когда поле low_delay = '1');
- время декодирования, рассчитанное из этого значения PTS (или из связанного с ним значения DTS), должно быть не раньше, чем время бесшовного декодирования;
- для элементарного видеопотока, если пакет сращивания не заканчивается полем sequence_end_code(), то "новый" поток должен начинаться с поля sequence_end_code(). После которого сразу следует заголовок sequence_header().

Теоретически, поскольку такие соединения приводят к нарушению непрерывности декодирования, они приводят в нарушению непрерывности воспроизведения модуля представления (т. е. промежутки "мертвого" времени переменной длины между воспроизведением двух последовательных видеокадров, между двумя последовательными аудиокадрами). На практике, результат будет зависеть от того, как реализован декодер, особенно, видеодекодер. В некоторых видеодекодерах предпочтительным решением может быть "стоп-кадр" в течение одного или нескольких видеокадров. См. Часть 4 стандарта ИСО/МЭК 13818.

K.2.2 При бесшовных объединениях

Цель отсутствия нарушений непрерывности декодирования заключается в том, чтобы не было нарушений непрерывности воспроизведения. В случае аудиопотока это всегда можно гарантировать. Но следует отметить, что в случае видеопотока для описанных ниже случаев 1) и 2) непрерывность воспроизведения, теоретически, невозможна:

1) "Старый" поток заканчивается в конце последовательности с малой задержкой, а "новый" поток начинается в начале последовательности с большой задержкой.
2) "Новый" поток заканчивается в конце последовательности с большой задержкой, и "новый" поток начинается в начале последовательности с малой задержкой.

Эффекты, вносимые такими ситуациями, зависят от варианта реализации. Например, для случая 1, может потребоваться воспроизводить видеокадр в течение двух периодов кадра, а для случая 2 может потребоваться пропустить кадр. Однако, технически возможно, чтобы некоторые варианты реализации поддерживали такие ситуации без каких-либо нежелательных эффектов.

Кроме того, в соответствии с § 6.1.1.6 Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2, поле sequence_end_code() должно присутствовать перед первым заголовком sequence_header() "нового" потока, если, как минимум, один параметр последовательности (т. е. параметр, определенный в заголовке последовательности или в расширении заголовка последовательности) имеет в обоих потоках различное значение, с единственным исключением, которое определено в матрице квантования. Например, если поле bit_rate имеет в "новом" потоке значение, отличающееся от его значения в "старом" потоке, то должно присутствовать поле sequence_end_code(). Таким образом, если пакет сращивания не заканчивается в поле sequence_end_code, то "новый" пакет должен начинаться с поля sequence_end_code, после которого следует заголовок sequence_header.

В соответствии с предыдущим параграфом, наличие поля sequence_end_code будет обязательным в большинстве объединений, даже в бесшовных. Следует заметить, что в Рекомендации МСЭ-Т H.262 | ИСО/МЭК 13818-2 определяется процесс декодирования видеопоследовательностей (т. е. данных, находящихся между полями sequence_header() и sequence_end_code()), и ничего не определено относительно того, как обрабатывать изменение последовательности. Таким образом, работа декодеров при наличии таких объединений описана в Части 4 стандарта ИСО/МЭК 13818.

K.2.3 Переполнение буфера

Даже если оба элементарных потока до объединения соответствовали модели T-STD, это вовсе не обязательно служит гарантией того, что объединенный поток не переполнит буферы STD через некоторое время, в течение которого в этих буферах находятся биты обоих потоков.

Для видеопотока с постоянной скоростью передачи, если "старому" потоку не применялись никакие особые условия, и если в процессе объединения не принималось никаких особых мер предосторожности, такое переполнение возможно в том случае, когда скорость передачи "нового" видеопотока больше, чем скорость передачи "старого" видеопотока. Тем не менее, несомненно, верно, что буферы MB_n и EB_n декодера T-STD не переполняются, если биты доставляются на вход T-STD со "старой" скоростью. Но, если скорость доставки на вход TB_n становится более высокой еще до того, как "старые" биты будут полностью удалены из T-STD, степень наполнения буферов STD станет больше, чем в случае, если бы "старый" поток продолжался без сращивания, и это может привести к переполнению буфера EB_n и/или MB_n. Для видеопотока с переменной скоростью передачи эта же проблема может возникнуть, если скорость доставки "нового" потока выше, чем скорость доставки, которая была предусмотрена во время создания "старого" потока. Такая ситуация является запрещенной.

Однако кодер может генерировать "старый" поток, для того чтобы добавить условия управления буфером VBV в окрестностях точек объединения, так, чтобы скорость любого "нового" видеопотока была бы меньше выбранного значения. Например, для точки бесшовного объединения, такие дополнительные условия могут быть указаны значением 'splice_type', данные которого соответствуют указанным в таблицах с 2-7 по 2-20 в разделе 'splice_decoding_delay' и 'max_splice_rate'. В таком случае, если скорость передачи видеопотока "нового" потока меньше значения 'max_splice_rate', гарантируется, что объединенный поток не приведет к переполнению буфера в течение времени, пока биты обоих потоков находятся в буфере декодера T-STD.

В том случае, когда такие ограничения не применяются, этой проблемы можно избежать, вводя в доставку битов между "старым" и "новым" потоками время простоя, для того чтобы дать возможность буферам T-STD достаточно опустеть перед доставкой битов "нового" потока. Если время, когда последний байт последнего модуля доступа "старого" потока поступает в STD, мы назовем t_{in}, а время, когда он покидает STD, мы назовем t_{out}, то достаточно гарантировать, что в T-TD более не поступает новых битов объединенного потока в течение интервала времени $[t_{in}, t_{out}]$, и "старый" поток продолжается без объединения. Например, в том случае, когда "старый поток" имеет постоянную скорость передачи R_{old}, а "новый" имеет постоянную скорость передачи R_{new}, то, во избежание риска переполнения буферов, достаточно ввести время простоя T_d, удовлетворяющее следующему соотношению:

$$T_d \geq 0 \text{ и } T_d \geq (t_{out} - t_{in}) \times (1 - R_{old}/R_{new})$$

Приложение L

Процедура регистрации (см. § 2.9)

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

L.1 Процедура запроса зарегистрированного идентификатора (RID)

Те, кто запрашивает зарегистрированный идентификатор (RID), должны обращаться в уполномоченный орган по регистрации. Формы заявлений также должны быть получены от уполномоченного органа по регистрации. Информация, которую должен предоставить заявитель, приводится в разделе L.3. Компании и организации имеют равное право заявления.

L.2 Область ответственности органа регистрации

В настоящем подразделе описывается основная область ответственности органа регистрации по регулированию регистрации идентификаторов авторского права (copyright_identifier); некоторые другие области ответственности описаны в Директивах JTC 1. Уполномоченный орган по регистрации должен:

a) реализовать процедуру регистрации по заявлению с запросом уникального RID в соответствии с Приложение H/Директивами JTC 1;

b) принимать и обрабатывать заявления на назначение идентификаторов типа произведения, поступающих из органов по регистрации авторского права;

c) проверить, какие из полученных заявлений соответствуют данной процедуре регистрации, и в течение 30 дней после их получения проинформировать заявителя о назначенном ему RID;

d) в течение 30 дней после получения заявления в письменной форме проинформировать тех заявителей, чьи заявления отклонены, и также проинформировать заявляющую сторону о возможности и правилах процесса апелляции;

e) поддерживать достоверный регистр назначенных значений RID. Должны допускаться и сохраняться в уполномоченном органе по регистрации изменения контактной информации и технических спецификаций;

f) обеспечивать доступность содержания регистра по запросу любой заинтересованной стороны;

g) поддерживать базы данных форм удовлетворенных и отклоненных запросов RID. Стороны, запрашивающие техническую информацию в формате конфиденциальной информации, которая имеет идентификатор copyright_identifier, должны получить доступ к такой информации, которая входит в состав базы данных, поддерживаемой уполномоченным органом по регистрации;

h) ежегодно, по взаимно согласованному графику, отчитываться о своих действиях перед JTC 1, ITTF и Секретариатом JTC 1/SC 29, или назначенными ими представителями.

L.2.1 Контактная информация уполномоченного органа по регистрации

Название организации:

Адрес:

Телефон :

Факс:

L.3 Область ответственности сторон, запрашивающих RID

Сторона, запрашивающая для целей идентификации авторского права, должна:

a) подать заявление, используя форму и процедуры, предоставленные уполномоченным органом по регистрации;

b) предоставить контактную информацию, описывающую, как может быть получено полное описание организации – владельца авторского права – на недискриминационной основе;

c) привести технические подробности синтаксиса и семантики формата данных, использованного для описания аудиовизуальных произведений или других произведений – объектов авторского права в поле **additional_copyright_info**. После того, как синтаксис, использованный для дополнительной информации об авторском праве, зарегистрирован, он не должен меняться;

d) согласиться начать использование назначенного идентификатора copyright_identifier в течение разумно-необходимого промежутка времени;

e) поддерживать постоянную регистрацию формы заявления и уведомлений, получаемых от уполномоченного органа по регистрации, для каждого назначенного идентификатора copyright_identifier.

L.4 Апелляционная процедура для отклоненных обращений

Сформирована Рабочая группа по регистрации (RMG), имеющая право рассматривать апелляции, связанные с отклоненными запросами RID. Членами RMG должны выбираться P и L членами технического комитета ИСО, ответственного за разработку настоящую Рекомендацию | Международный стандарт. У группы должен быть руководитель и секретариат, выбранные из ее членов. Уполномоченный орган по регистрации имеет право назначить в эту группу одного наблюдателя без права голосования.

Область ответственности RMG должна быть следующей:

a) пересматривать и выполнять необходимые действия для всех апелляций в течение разумно необходимого периода времени;

b) в письменной форме информировать организации, подавшие апелляции о повторном рассмотрении их заявления, о мнении RMG по данному вопросу;

c) рассматривать ежегодный отчет о действиях уполномоченного органа по регистрации;

d) предоставлять членам организаций ИСО информацию, касающуюся сферы действия уполномоченного органа по регистрации.

Приложение M

Форма заявления на регистрацию (см. § 2.9)

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

M.1 Контактная информация об организации, запрашивающей зарегистрированный идентификатор (RID)

Название организации:

Адрес:

Телефон:

Факс:

e-mail:

M.2 Заявление о намерении применить назначенный RID

Область применения RID: используя руководящие указания, предоставленные уполномоченным органом по регистрации.

M.3 Дата предположительного применения RID

M.4 Уполномоченный представитель

Имя:

Должность:

Адрес:

Подпись: _____

M.5 Только для официального использования органом по регистрации

В регистрации отказано: _____ Причина отказа в регистрации: Регистрация выполнена: _____ Регистрационный номер: _____

Приложение 1 – Приложение технических подробностей запрашиваемого формата данных.

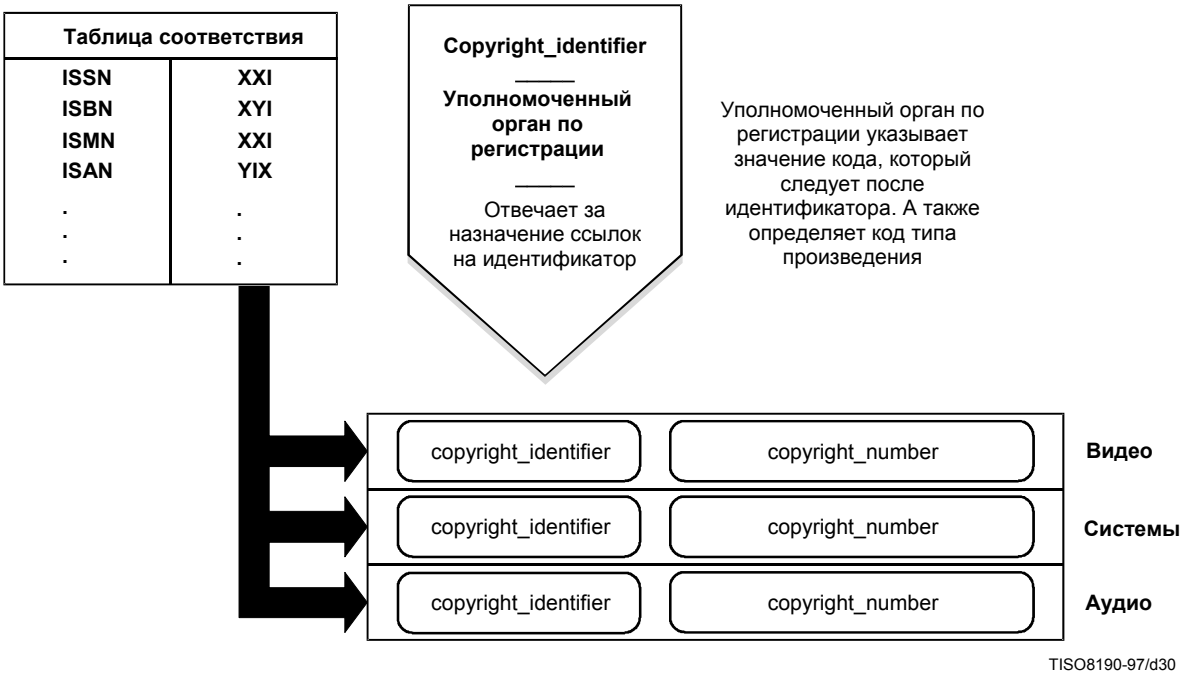
Приложение 2 – Приложение уведомления о процедуре апелляции для отклоненных заявлений.

Приложение N

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Уполномоченный орган по регистрации

Диаграмма регистрационной структуры (см. § 2.9)



Примеры

Все идентификаторы copyright_identifier регистрируются уполномоченным органом по регистрации с уникальными номерами copyright_number, стандартизованными ИСО. Каждая организация, которая назначает номера copyright_number, запрашивает у уполномоченного органа по регистрации конкретный copyright_identifier, например, библиотека Staatsbibliothek Preussischer Kulturbezits, назначенная ИСО регулировать номера ISBN, запрашивает у уполномоченного органа по регистрации определенный copyright_identifier для нумерации книг

Приложение O

Процедура регистрации (см. § 2.10)

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

O.1 Процедура запроса RID

Те, кто запрашивает зарегистрированный идентификатор (RID), должны обращаться в уполномоченный орган по регистрации. Формы заявлений должны быть получены от уполномоченного органа по регистрации. Заявитель должен предоставить информацию, определенную в § O.4. Компании и организации имеют равное право заявления.

O.2 Область ответственности органа регистрации

В настоящем подразделе описывается основная область ответственности органа регистрации по регулированию регистрации идентификаторов формата конфиденциальных данных format_identifier; некоторые другие области ответственности описаны в Директивах JTC 1. Уполномоченный орган по регистрации должен:

a) реализовать процедуру регистрации по заявлению с запросом уникального RID в соответствии с Приложение H/Директивами JTC 1;

b) принимать и обрабатывать заявления на назначение идентификаторов, поступающих от заявителей;

c) проверить, какие из полученных заявлений соответствуют данной процедуре регистрации, и в течение 30 дней после их получения проинформировать заявителя о назначенном ему RID;

d) в течение 30 дней после получения заявления в письменной форме проинформировать тех заявителей, чьи заявления отклонены, и порекомендовать им своевременно рассмотреть вопрос о повторной подаче заявления;

e) поддерживать достоверный регистр назначенных идентификаторов. Должны допускаться и сохраняться в уполномоченном органе по регистрации изменения контактной информации и технических спецификаций;

f) обеспечивать доступность содержания регистра по запросу национальных организаций JTC 1, которые являются членами ИСО или МЭК, обеспечивать взаимодействие ИСО или МЭК и любых заинтересованных сторон;

g) поддерживать базы данных форм удовлетворенных и отклоненных запросов RID. Стороны, запрашивающие техническую информацию в формате конфиденциальной информации, которая имеет идентификатор RID, должны иметь доступ к такой информации, которая входит в состав базы данных, поддерживаемой уполномоченным органом по регистрации;

h) ежегодно, по взаимно согласованному графику, отчитываться о своих действиях перед JTC 1, ITTF и Секретариатом JTC 1/SC 29;

i) везде, где возможно допускать применение существующих RID.

O.3 Контактная информация органа регистрации

O.4 Область ответственности сторон, запрашивающих RID

Сторона, запрашивающая идентификатора format_identifier, должна:

a) подать заявление, используя форму и процедуры, предоставленные уполномоченным органом по регистрации;

b) предоставить описание предназначения регистрируемого двоичного потока, и необходимые технические подробности, указанные в форме заявления;

c) предоставить контактную информацию, описывающую, как может быть получено полное описание на недискриминационной основе;

d) согласиться начать использование назначенного идентификатора RID в течение разумно-необходимого промежутка времени;

e) поддерживать постоянную регистрацию формы заявления и уведомлений, получаемых от уполномоченного органа по регистрации, для назначенного идентификатора RID.

O.5 Апелляционная процедура для отклоненных обращений

Сформирована Рабочая группа по регистрации (RMG), имеющая право рассматривать апелляции, связанные с отклоненными запросами RID. Членами RMG должны выбираться P и L членами технического комитета ИСО, ответственного за настоящую Спецификацию. У группы должен быть руководитель и секретариат, выбранные из ее членов. Уполномоченный орган по регистрации имеет право назначить в эту группу одного наблюдателя без права голосования.

Область ответственности RMG должна быть следующей:

a) пересматривать и выполнять необходимые действия для всех апелляций в течение разумно необходимого периода времени;

b) в письменной форме информировать организации, подавшие апелляции о повторном рассмотрении их заявления, о мнении RMG по данному вопросу;

c) рассматривать ежегодный отчет о действиях уполномоченного органа по регистрации;

d) предоставлять членам организаций ИСО и национальных комитетов МЭК информацию, касающуюся сферы действия уполномоченного органа по регистрации.

Приложение P

Форма заявления на регистрацию

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

P.1 Контактная информация об организации, запрашивающей RID

Название организации:

Адрес:

Телефон:

Факс:

e-mail:

телекс:

P.2 Запрос конкретного RID

ПРИМЕЧАНИЕ. – Если система уже реализована и используется, заполните этот пункт, а также пункт P.3, затем пропустите пункт P.6; в ином случае оставьте это поле пустым и переходите к P.4.

P.3 Краткое описание RID, которое используется в развертываемой системе

P.4 Заявление о намерении применить назначенный RID

P.5 Дата предположительного применения RID

P.6 Уполномоченный представитель

Имя:

Должность:

Адрес:

Подпись: _____

P.7 Для официального использования органом по регистрации

В регистрации отказано: _____ Причина отказа в регистрации: Регистрация выполнена: _____ Регистрационный номер: _____

Приложение 1. – Приложение технических подробностей запрашиваемого формата данных.

Приложение 2. – Приложение уведомления о процедуре апелляции для отклоненных заявлений.

Приложение Q

Модели буфера T-STD и P-STD для ADTS, соответствующего ИСО/МЭК 13818-7

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Q.1 Введение

Модель декодера конечной системы Транспортного потока для аудиопотоков определяется в § 2.4.2. В настоящем Приложении описывается модель буфера для ADTS, соответствующего ИСО/МЭК 13818-7 ADTS.

Аудиопотоки ADTS, соответствующие ИСО/МЭК 13818-7, можно распознать в мультиплексированном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1, по наличию идентификаторов потока stream_id=0x110yyyyy ('y' = "не обращайте внимания") и stream_type=0x0F, как определено в таблицах 2-22 и 2-23.

Q.2 Интенсивность утечки из транспортного буфера

Для аудиоданных, кроме аудиопотоков ADTS, соответствующих ИСО/МЭК 13818-7, интенсивность утечки из транспортного буфера равна 2 Мбит/с. Однако, эта скорость меньше, чем максимальная скорость для ADTS, соответствующих ИСО/МЭК 13818-7. Следовательно, скорость утечки для потока ADTS, соответствующего ИСО/МЭК 13818-7, устанавливается в значение, отличное от значения для аудиопотоков, соответствующих ИСО/МЭК 11172-3 и ИСО/МЭК 13818-3.

Элементарный поток ADTS, соответствующий ИСО/МЭК 13818-7, состоит из одного или нескольких каналов. Максимальная скорость каждого канала равна 576 кбит/с, где частота дискретизации равна 96 кГц. Следовательно, скорость утечки для ADTS, соответствующих ИСО/МЭК 13818-7, рассчитывается в соответствии со следующим уравнением.

$$Rx_n = 1{,}2 \times R_{max} \times N \text{ битов в секунду,}$$

где:

R_{max} имеет постоянное значение 576 кбит/с, определенное в п. 3.2.2 стандарта ИСО/МЭК 13818-7. Это верхняя граница скорости передачи в канале AAC потока ADTS в соответствии с максимальным значением частоты дискретизации (т. е. Fs = 96 кГц),

и где:

N – количество аудиоканалов, для которых необходим отдельный буфер декодера, в этом элементарном потоке (т. е. отдельные канальные потоки на одном элементе канала или паре элементов канала и независимо коммутируемых элементах канала).

Q.3 Размер буфера

Для аудиоданных, кроме аудиопотоков ADTS, соответствующих ИСО/МЭК 13818-7, размер основного буфера равен 3584 байтов. Это размер, однако, меньше, чем максимальный размер входного буфера декодера для ADTS, соответствующих ИСО/МЭК 13818-7. Следовательно, размер основного буфера для потока ADTS, соответствующего ИСО/МЭК 13818-7, устанавливается в значение, отличное от значения для аудиопотоков, соответствующих ИСО/МЭК 11172-3 и ИСО/МЭК 13818-3.

Размер основного буфера для ADTS, соответствующих ИСО/МЭК 13818-7 13818-7, рассчитывается следующим образом:

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh},$$

где BS_{oh} – размер буфера заголовка пакета PES определяется следующим образом:

$$BS_{oh} = 528 \text{ байтов}$$

и BS_{mux} – размер буфера дополнительного мультиплексирования определяется следующим образом:

$$BS_{mux} = 0{,}004 \text{ секунд} \times R_{max} \times N$$

и BS_{dec} – размер буфера модуля доступа определяется следующим образом:

$$BS_{dec} = 6144 \text{ битов} \times N,$$

где:

R_{max} имеет постоянное значение 576 кбит/с, определенное в п. 3.2.2 стандарта ИСО/МЭК 13818-7. Это верхняя граница скорости передачи в канале AAC потока ADTS в соответствии с максимальным значением частоты дискретизации (т. е. Fs = 96 кГц),

и где:

N – количество аудиоканалов, для которых необходим отдельный буфер декодера, в этом элементарном потоке (т. е. отдельные канальные потоки на одном элементе канала или паре элементов канала и независимо коммутируемых элементах канала).

Q.3.1 TBS_n: такой же, как и для других аудиопотоков

Нет никаких различий в размере сглаживающего буфера TB_n для потоков ADTS, соответствующих ИСО/МЭК 13818-7, и для других аудиопотоков. Следовательно, нет необходимости менять TBS_n, размер которого равен TB_n.

Q.3.2 BS_{mux}: отличается от других аудиопотоков

Размер буфера для дополнительного мультиплексирования (BS_{mux}) должен быть изменен таким образом, чтобы учитывалась нестабильность задержки величиной до 4 мс. Этот подход аналогичен подходу, принятому для других потоков, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1.

Q.3.3 BS_{dec}: отличается от других аудиопотоков

Размер буфера модуля доступа (BS_{dec}) основан на размере входного буфера декодера элементарного потока. Как определено в п. 3.2.2 стандарта ИСО/МЭК 13818-7, общий размер входного буфера декодера равен 6144 бита, умноженные на количество каналов, которое требуется для каждого входного буфера декодера.

Q.3.4 BS_{oh}: отличается от других аудиопотоков

BS_{oh} соответствует размеру дополнительных данных в заголовке пакета PES.

В § 2.4.2.6 сказано:

> *Задержка любых данных, передаваемых через буферы конечных декодеров системы, должна быть меньше или равна одной секунде, за исключением случаев передачи данных неподвижных изображений.*

Кроме того, в § 2.7.4 сказано:

> *Программный поток и Транспортный поток должны быть созданы таким образом, чтобы максимальная разница между кодированными метками времени воспроизведения, относящимися к каждому видео или аудиопотоку, составляла 0,7 секунды.*

Размер BS_{oh} должен быть установлен в соответствии с размером дополнительных данных в заголовке пакета PES, когда поток AAC пакетируется согласно вышеприведенным правилам. Максимальный размер заголовка пакета PES равен 264 байтов. Следовательно, BS_{oh} = 528 байтов, т. е. вдвое больше максимального размера заголовка пакета PES, что гарантирует такое положение дел, при котором в основной буфер может поместиться, как минимум, два заголовка пакета PES, вне зависимости от размера заголовка пакета PES. Это означает, что заголовки пакета PES с меткой PTS могут быть размещены с интервалом менее 0.7 секунд, даже, когда в основном буфере будут находиться данные продолжительностью 1 секунда.

Пример: частота дискретизации = 48 кГц

Размер заголовка пакета PES без каких-либо дополнительных полей, за исключением PTS, равен 18 байтов. Количество модулей доступа в течение одной секунды составляет примерно 47. Когда в основном буфере находятся данные продолжительностью 1 секунда (т. е. в наихудшем случае), Размер дополнительных данных в заголовке пакета PES может соответствовать размеру BS_{oh}, если в одном пакете находится два и более модуля доступа.

number_of_AU = 48 кГц/1024 = 46 875 в секунду

(number_of_AU/2) × 18[байтов] = 421 875 байтов < BS_{oh}

Чаще размер заголовка пакета PES может соответствовать размеру BS_{oh}, если задержка любых данных, передаваемых через основной буфер, меньше одной секунды.

Q.4 Вывод

Буфер модели декодера должен соответствовать максимальному размеру буфера; однако, поток AAC может поддерживать до 48 каналов с очень высокой скоростью. Следовательно, для определения скорости утечки размера основного буфера используется 3 уровня значений количества каналов, 2, 8 и 48. В случае 2 каналов, для поддержания совместимости используются та же самая скорость утечки и размер основного буфера, то и для обычных значений. В других случаях (8 и 48) применяются предложенные формулы.

Скорость утечки в декодере T-STD для потоков ADTS, соответствующих ИСО/МЭК 13818-7,

Количество каналов	**Rx_n [бит/с]**
1-2	2 000 000
3-8	5 529 600
9-12	8 294 400
13-48	33 177 600

Каналы: Количество выходных аудиоканалов с полной шириной полосы пропускания плюс количество независимо коммутируемых элементов соединения каналов в отдельном элементарном аудиопотоке. Например, в типичном случае, когда независимо коммутируемые элементы соединения каналов отсутствуют, для передачи сигнала моно используется 1 канал, для стерео – 2 канала, а для передачи 5.1 канала объемного звучания – 5 каналов (канал низкочастотных эффектов (LFE) не учитывается).

Размер основного буфера декодера T-STD для потоков ADTS, соответствующих ИСО/МЭК 13818-7

Количество каналов	**BS_n [байтов]**
1-2	3 584
3-8	8 976
9-12	12 804
13-48	51 216

Каналы: Количество выходных аудиоканалов с полной шириной полосы пропускания плюс количество независимо коммутируемых элементов соединения каналов в отдельном элементарном аудиопотоке. Например, в типичном случае, когда независимо коммутируемые элементы соединения каналов отсутствуют, для передачи сигнала моно используется 1 канал, для стерео – 2 канала, а для передачи 5.1 канала объемного звучания – 5 каналов (канал низкочастотных эффектов (LFE) не учитывается).

Для Программного потока вышеуказанный размер основного буфера должен быть установлен в поле P-STD_buffer_scale и P-STD_buffer_size следующим образом.

Количество каналов	**P-STD_buffer_scale**	**P-STD_buffer_size**
1-2	0	28
3-8	0	71
9-48	0	401

Приложение R

Передача сцен, соответствующих ИСО/МЭК 14496, в потоках, соответствующих Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1

(Данное Приложение не является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

R.1 Процедура доступа к содержанию для компонентов программы, соответствующих ИСО/МЭК 14496, в Программном потоке

В последующем разделе представлена эталонная процедура испытаний в приемнике для оценки компонентов программы, соответствующих ИСО/МЭК 14496, передаваемых в Программном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1. Здесь предполагается, что Программный поток включает в себя карту Программного потока (передаваемую в пакете PES, имеющем идентификатор stream_id = 0xBC):

- Получить карту Программного потока.
- Идентифицировать дескриптор IOD в первой цепочке дескрипторов.
- Определить идентификаторы элементарных потоков (ES_ID) дескриптора объекта, описания сцены и других потоков, описанных в дескрипторе исходного объекта.
- Получить дескриптор SL и дескриптор FMC во второй цепочке дескрипторов для полей elementary_stream_id = 0xFA и 0xFB, в зависимости от конкретного случая.
- Создать из этих дескрипторов таблицу преобразования потока между идентификаторами ES_ID и соответствующими полями elementary_stream_id плюс, при необходимости, канал FlexMux.
- Определить местоположение потока дескрипторов объекта, используя его идентификатор ES_ID и таблицу преобразования потока.
- Определить местоположение других потоков, описанных в дескрипторе исходного объекта, используя их идентификаторы ES_ID и таблицу преобразования потока.
- Непрерывно контролировать поток дескрипторов объекта и идентифицировать идентификаторы элементарных потоков (ES_ID) для дополнительных потоков.
- Определить местоположение дополнительных потоков, используя их идентификаторы ES_ID и таблицу преобразования потока.

На рисунке R.1 показан пример содержания Программного потока, соответствующего ИСО/МЭК 14496, состоящего из потока дескрипторов объекта, потока описания сцены (BIFS-Command), потока BIFS-Anim и потока IPMP. Все потоки, соответствующие ИСО/МЭК 14496, мультиплексируются в отдельный поток FlexMux.

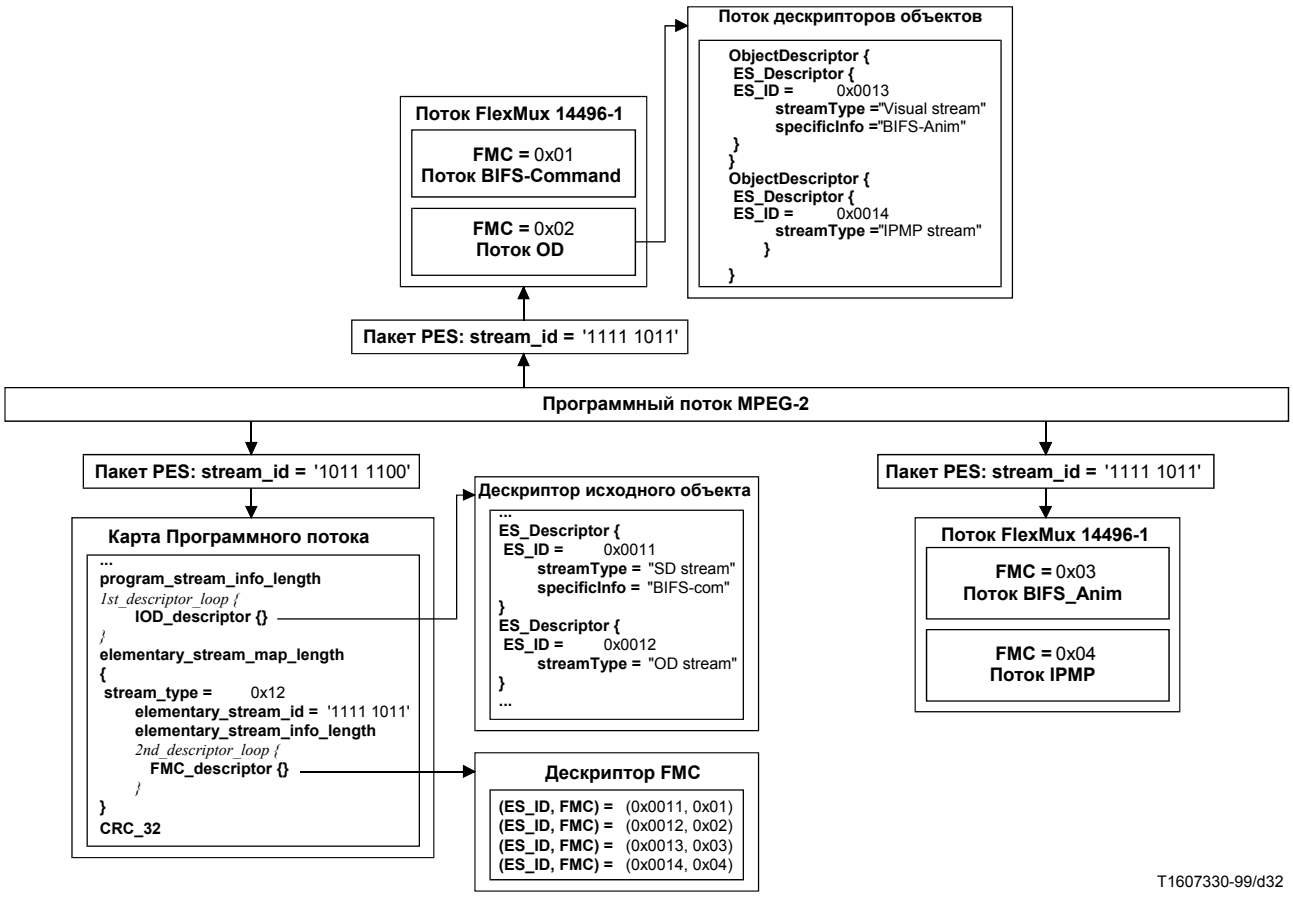


Рисунок R.1 – Пример передачи содержания, соответствующего ИСО/МЭК 14496, в Программном потоке

R.2 Процедура доступа к содержанию для компонентов программы, соответствующих ИСО/МЭК 14496, в Транспортном потоке

В последующем разделе представлена эталонная процедура испытаний в приемнике для оценки компонентов программы, соответствующих ИСО/МЭК 14496, передаваемых в транспортном потоке, соответствующем Рекомендации МСЭ-Т H.222.0 | ИСО/МЭК 13818-1:

- Получить Таблицу преобразования программы для желаемой программы.
- Идентифицировать дескриптор IOD в первой цепочке дескрипторов.
- Определить идентификаторы элементарных потоков (ES_ID) дескриптора объекта, описания сцены и других потоков, описанных в дескрипторе исходного объекта.
- Получить набор всех дескрипторов SL и дескрипторов FMC во второй цепочке дескрипторов для любых идентификаторов elementary_PID.
- Создать из этих дескрипторов таблицу преобразования потока между идентификаторами ES_ID и соответствующими полями elementary_PID плюс, при необходимости, канал FlexMux.
- Определить местоположение потока дескрипторов объекта, используя его идентификатор ES_ID и таблицу преобразования потока.
- Определить местоположение других потоков, описанных в дескрипторе исходного объекта, используя их идентификаторы ES_ID и таблицу преобразования потока.
- Непрерывно контролировать поток дескрипторов объекта и идентифицировать идентификаторы элементарных потоков (ES_ID) для дополнительных потоков.
- Определить местоположение дополнительных потоков, используя их идентификаторы ES_ID и таблицу преобразования потока.

На рисунке R.2 показан пример передачи элементов программы в Транспортном потоке, соответствующем ИСО/МЭК 14496, состоящем из потока дескрипторов объекта, потока описания сцены (BIFS-Command), элементарных потоков BIFS-Anim и IPMP. Потоки BIFS-Command и OD передаются в секциях ISO_IEC_14496_section, тогда как элементарные потоки BIFS-Anim и IPMP передаются в пакетах PES, обозначенных отличительными значениями elementary_PID, без применения функций потока FlexMux, соответствующего ИСО/МЭК 14496-1.

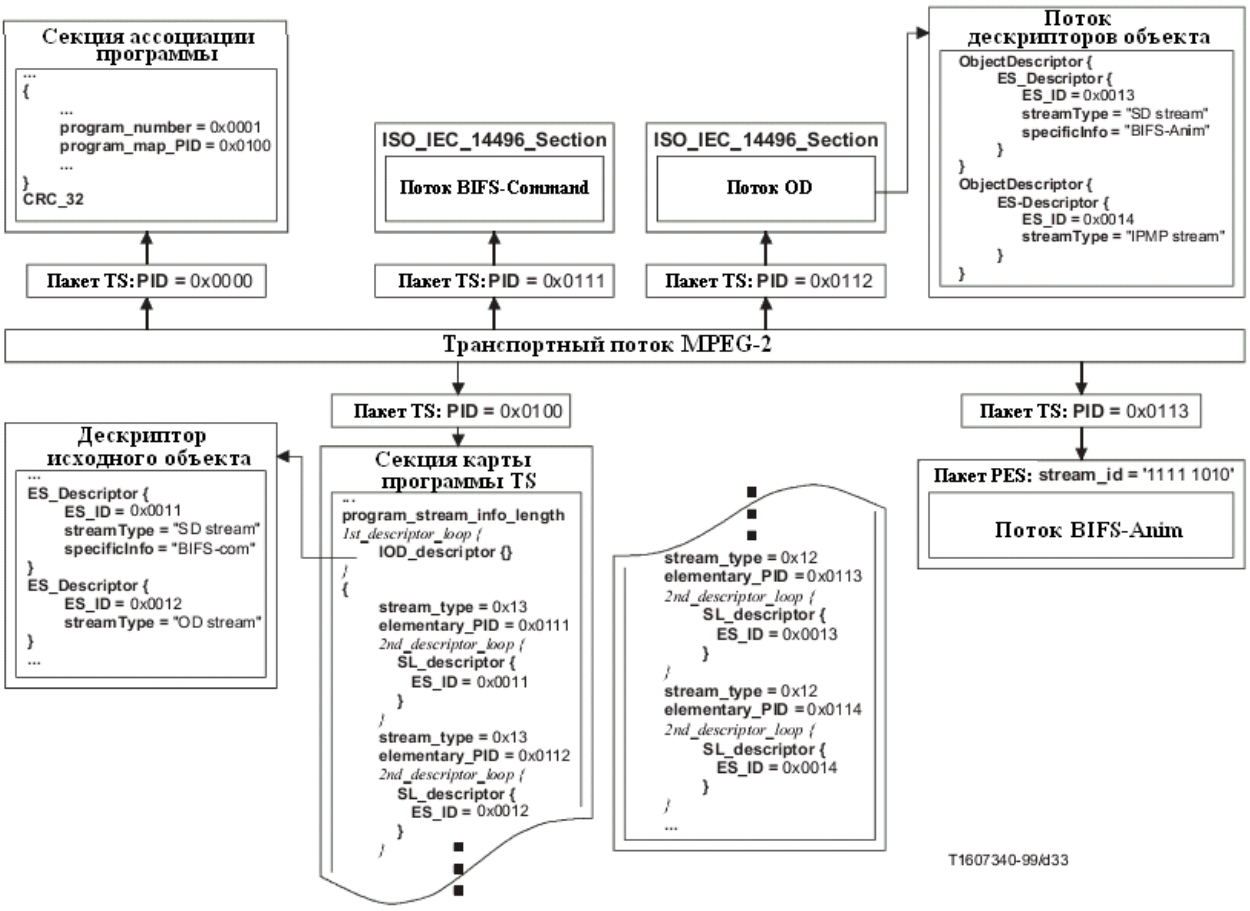


Рисунок R.2 – Пример передачи содержания, соответствующего ИСО/МЭК 14496, в Транспортном потоке

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия A	Организация работы МСЭ-Т
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	**Аудиовизуальные и мультимедийные системы**
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	Управление электросвязью, включая СУЭ и техническое обслуживание сетей
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных, взаимосвязь открытых систем и безопасность
Серия Y	Глобальная информационная инфраструктура, аспекты протокола Интернет и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи

Отпечатано в Швейцарии
Женева, 2010 г.